



Squelettisation en un balayage. Application à la caractérisation osseuse.

Aurore Arlicot

► To cite this version:

Aurore Arlicot. Squelettisation en un balayage. Application à la caractérisation osseuse.. Imagerie médicale. L'Université Nantes Angers Le Mans, 2012. Français. NNT : . tel-00781222

HAL Id: tel-00781222

<https://theses.hal.science/tel-00781222>

Submitted on 25 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse de Doctorat

Aurore Arlicot

*Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le label de l'Université de Nantes Angers Le Mans*

Discipline : Imagerie

Spécialité : Automatique Informatique Appliquée

Laboratoire : Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN)

Soutenue le 24 octobre 2012

École doctorale : 503 (STIM)

Thèse n° : ED 503-178

Squelettisation en un balayage.
Application à la caractérisation osseuse.

JURY

Rapporteurs :	M. Michel COUPRIE , Professeur, ESIEE, Université Paris-Est M. Olivier DÉFORGES , Professeur, INSA de Rennes
Examineurs :	M. Christophe LOHOU , Professeur, IUT Puy en Velay, Université d'Auvergne Clermont 1 M. Jean Michel BOULER , Professeur, Odontologie, Université de Nantes
Invité :	M. Yves AMOURIQ , Professeur, Odontologie, Université de Nantes
Directeur de thèse :	M. Jeanpierre GUÉDON , Professeur, Polytech, Université de Nantes
Co-directeur de thèse :	M. Nicolas NORMAND , Maître de conférences, Polytech, Université de Nantes

Remerciements

[illegible]

Table des matières

Introduction générale	11
I Géométrie discrète	13
Introduction	15
1 L'espace discret et distances discrètes	17
1.1 Introduction	17
1.2 Voisinage et connexité	18
1.3 Notion de topologie d'une image	20
1.3.1 Les composantes connexes	21
1.3.2 Le nombre d'Euler	23
1.3.3 Points simples	24
1.4 Les distances discrètes	28
1.4.1 Définition	28
1.4.2 Distances simples	30
1.4.3 Distances à séquence de voisinage	31
1.4.4 Distances de chanfrein	35
1.5 Transformation en distances	38
1.6 Axe médian	42
1.7 Propagation de l'information	45
1.8 Conclusion	47
2 La squelettisation	49
2.1 Introduction	49
2.2 Le squelette et ses propriétés	50
2.3 Squelettisation basée sur l'amincissement (Thinning)	51
2.4 Squelettisation basée sur les distances	53
2.5 Squelettisation par approche hybride	56
2.6 Conclusion	60

Conclusion	61
II La squelettisation en un balayage	63
Introduction	65
3 Calcul de la carte de distance translatée	69
3.1 Introduction	69
3.2 Distances à poids variables	70
3.3 Calcul de DT'_X	71
3.3.1 Translation des centres des disques	71
3.3.2 La séquence de voisinage	73
3.3.3 Algorithme et exemple	78
3.4 Équivalence et réversibilité de DT_X et DT'_X	80
3.4.1 Équivalence entre DT_X et DT'_X	81
3.4.2 Réversibilité : Calcul de DT_X à partir de DT'_X	82
3.5 Résultats	83
3.6 Conclusion	90
4 L'axe médian	93
4.1 Introduction	93
4.2 Calcul de l'axe médian	94
4.3 Résultats	97
4.4 Conclusion	103
5 La squelettisation	105
5.1 Introduction	105
5.2 Algorithme de référence	106
5.3 De rlk vers $l'k'r$	111
5.3.1 Algorithme ordonné en ligne/rayon de disques/colonne ($l'rk'$)	111
5.3.2 Algorithme ordonné en ligne/colonne/rayon de disques ($l'k'r$)	114
5.4 Résultats	123
5.4.1 Augmentation de la résolution de l'image	125
5.4.2 Dernier amincissement	131

5.4.3	Temps de calcul	133
5.4.4	Discussion	136
5.5	Conclusion	142
Conclusion		145
 III Application biomédicale		 149
Introduction		151
6	Étude de l'os trabéculaire	153
6.1	Qu'est-ce que l'os trabéculaire?	153
6.2	Comment caractériser l'os?	155
6.3	Acquisition et pré-traitements	160
6.4	Conclusion	167
7	Extraction des caractéristiques de l'os trabéculaire	169
7.1	Définition des caractéristiques à extraire	169
7.1.1	Les points terminaux	170
7.1.2	Les points d'intersections	172
7.1.3	La longueur des trabécules	173
7.1.4	La largeur des trabécules	174
7.2	Méthode d'extraction des informations	176
7.2.1	Rappel des étapes précédentes	177
7.2.2	Dernier amincissement	177
7.2.3	Extraction des points spéciaux	178
7.2.4	Étiquetage des trabécules	181
7.3	Résultats	187
Conclusion		191

Conclusion générale	195
Perspectives	201
1 Squelettisation 3D en un balayage	201
2 Distances utilisées	202
3 Amélioration de la méthode d'extraction des caractéristiques os- seuses	202
4 Autres applications	204
 IV Annexe	 205
1 Temps d'exécution	207
1.1 La base d'images testées	207
1.2 Carte de distance	209
1.3 Axe médian	212
1.4 Squelettisation	216
2 Extraction des caractéristiques osseuses	228
2.1 Caractéristiques extraites d'une image d'os.	229
2.2 Caractéristiques extraites par le logiciel d'odonto	238
 Production scientifique	 243
 Bibliographie	 245

Introduction générale

“La médecine est une science des pannes, celles de l’organisme humain... Mais si le médecin est un dépanneur - rien de plus, rien de moins - il est le dépanneur d’une machine dont il ne possède pas les plans.” Lucien Israël.

Les gens tombent malades et les médecins soignent les malades. Pour cela ils ont recours à différents outils de travail. L’imagerie médicale est notamment devenue un outil très précieux pour les médecins. Elle permet aux médecins de visualiser et d’étudier un corps sain et en déduire les différences avec ou sans pathologie donnée.

Il existe différents types d’imagerie médicale : les rayons X, l’imagerie par résonance magnétique (IRM), l’échographie,...

Chacune de ces modalités produit soit des images de projection comme la radiographie à rayons X, soit des coupes après reconstruction tomographique comme l’IRM.

À partir des images produites, les médecins peuvent visualiser un état physiologique on l’appelle “imagerie structurelle”. Elle donne des informations sur l’anatomie des organes comme la taille ou la forme. Les médecins peuvent également avoir recours à “l’imagerie fonctionnelle”. Comme son nom l’indique, ce type d’imagerie permet aux médecins de visualiser le fonctionnement d’un organe.

C’est dans ce contexte que se positionne ce travail : caractériser des formes soit d’organes soit de matériaux constituant d’organes. Les mesures s’effectuent sur des formes : c’est la première étape dite de « reconnaissance des formes ». Si on image l’ensemble d’un organe, on peut donc comparer statistiquement les paramètres explicitant la forme de l’organe vis-à-vis d’un ensemble d’organes sains ou atteints d’une maladie spécifique. C’est la seconde étape de « classification de formes ». On pourra ainsi classer l’organe du patient dans une classe saine/malade en fonction des paramètres de la forme. Si on image une partie d’organe constituée d’un ensemble de petites formes, le même raisonnement s’applique en réalisant des moyennes sur chaque élément de forme élémentaire qu’on est capable de dissocier des voisins.

Dans notre étude, nous visons à caractériser l’architecture osseuse ou plus précisément l’os trabéculaire. L’os trabéculaire a une architecture particulièrement compliquée (forme spongieuse), nous allons mettre en place des traitements d’image

adaptés.

Les images que nous allons analyser se composent de millions de pixels dont les valeurs sont comprises entre 0 et 255 (images dites en niveaux de gris). Les post-traitements qui ont été appliqués antérieurement à ce travail permettent d'obtenir une image binaire (dont la valeur des pixels est soit de 0 soit de 1) et réduisent la taille des images à un millier de pixels à traiter.

Les traitements d'images que nous souhaitons appliquer nécessitent le stockage en mémoire de $N \times N$ pixels constituant l'image.

Dans ce travail, nous réduisons l'occupation mémoire de ces traitements à $2N$ pixels. Outre cette meilleure gestion de la mémoire, nous remarquerons l'efficacité en terme de temps de calculs et ce, sans changer la complexité algorithmique de nos traitements.

La première partie de ce travail consiste à présenter l'espace dans lequel nous allons travailler et présenter les traitements d'images que nous allons développer.

La seconde partie présente les différentes phases du traitement que nous avons mis en œuvre.

La troisième partie montre comment ce traitement a été appliqué à la caractérisation de l'os.

Première partie

Géométrie discrète

Introduction

La géométrie discrète est un sous ensemble de la géométrie au même titre que la géométrie “continue”. Cette dernière nous a tous été enseignée dans notre enfance. Si la géométrie “continue” a pris le monopole de l’appellation géométrie, la géométrie discrète n’en n’est pas moins intéressante.

L’ère du numérique a véritablement propulsé la géométrie discrète au devant de la scène en mettant en cause “*le monde continu*”. Par effet boule de neige, la discrétisation de l’information nous a forcé à savoir analyser ces données discrètes.

L’utilisation de capteurs, lors de l’acquisition de l’image, provoque une discrétisation du monde réel. Si nous savions calculer la distance entre deux objets posés sur une table, il était par contre plus difficile d’estimer la distance entre les deux objets à partir d’une photographie de cette même scène. Car au delà du problème d’échelle de l’image de la scène, l’image est composée de pixels dû au mécanisme d’acquisition de celle ci.

C’est dans les années 1960 que des chercheurs français découvrent la morphologie mathématique, une discipline de l’analyse d’image. Il a alors fallu apprivoiser ce monde discret en commençant par définir l’espace discret. Des notions de connexités de pixels ont permis d’effectuer des liens entre pixels, qui aboutiront progressivement à la naissance de premières distances discrètes.

Cette première partie présente cet espace discret et certaines distances discrètes. La suite de ce travail concerne un descripteur de formes morphologiques très utilisé en analyse d’image : le squelette. Le processus de squelettisation est également présenté dans cette partie.

L'espace discret et distances discrètes

1.1 Introduction

Une image numérique peut être représentée comme un tableau $2D$ où chaque case du tableau représente un pixel et contient une valeur (0 ou 1 pour une image binaire et de 0 à 255 pour une image en niveau de gris). En $3D$ l'image est en fait un volume d'images qui est représenté par un tableau à 3 dimensions. Dans ce sens nous travaillerons, dans cette thèse, dans un espace discret composé de pixels. Cet espace discret est en fait un espace continu partitionné par un pavage régulier. Le pavage est une représentation de l'espace utilisé en imagerie numérique. Pour faire ce pavage régulier on utilise la même forme que l'on déplace pour remplir le plan. Il existe plusieurs types de pavages (voir figure 1.1). Parmi ces pavages le pavage carré reste le plus utilisé parce qu'il facilite l'extraction des points discrets. Ainsi chaque point a une paire de coordonnées correspondant à l'indice des lignes et des colonnes de l'image et en $3D$ chaque voxel a ainsi un triplet de coordonnées (la dernière coordonnée représentant le numéro de plan de l'image). Le centre de chaque cellule de ce pavage est un point discret. Le maillage est un graphe reliant les points discrets de chaque cellule voisine comme illustré sur la figure 1.1.

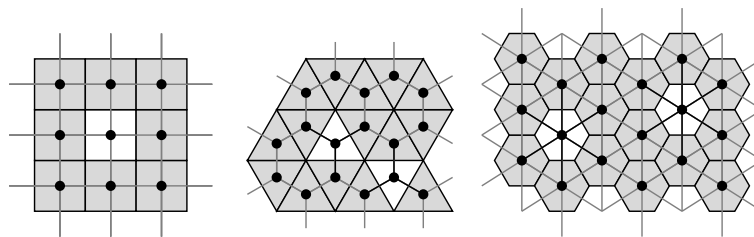


FIGURE 1.1 – Pavages associés à leur maillage carré, triangulaire et hexagonal

En $3D$ les pavages réguliers peuvent avoir des formes plus complexes. La forme du pixel, ou du voxel en $3D$, dépend de différentes contraintes :

- Acquisition de l'information ou l'étape de numérisation,
- Restitution de l'information (aspect visuel),
- Manipulation de l'information. Par exemple, un pixel triangulaire nécessite un codage particulier pour accéder à ses coordonnées.

Dans ce travail de thèse nous utiliserons des pixels carrés car il est plus pratique de manipuler un échantillonnage orthogonal.

Dans ce chapitre sont introduites des notions de base utilisées dans la suite de ce travail. Dans la section 1.2 nous décrivons comment deux points sont voisins puis connexes. Ces notions sont ensuite utilisées pour en déduire un chemin qui permet de décrire une forme. La définition et la caractérisation d'un point simple sont présentées dans la section 1.3.3. La section 1.4 définit ce qu'est une distance, quelles sont ses propriétés et présente ses principales mises en œuvre.

Enfin dans la dernière section 1.7 est présenté le problème général du déplacement dans un espace discret $2D$.

1.2 Voisinage et connexité

On considère un objet (ou une forme) X dans \mathbb{Z}^2 , un ensemble de points. Cet objet a une topologie et les points le constituant, des relations. Cette section présente quelques définitions de relations entre pixels.

Avec un échantillonnage carré, il existe plusieurs types de voisinages, les plus courants sont le 4-voisinage et le 8-voisinage. On les appelle aussi voisinages de

type 1 (\mathcal{N}_1) et de type 2 (\mathcal{N}_2) respectivement. Dans ce travail, nous continuerons à les noter de cette façon.

Définition 1 (Voisinage $\mathcal{N}_1(p)$). *Soit p un point de coordonnées (x_p, y_p) . Le voisinage de type 1 du point p (noté $\mathcal{N}_1(p)$) est l'ensemble des points q de coordonnées (x_q, y_q) qui vérifient :*

$$|x_p - x_q| + |y_p - y_q| \leq 1 \quad (1.1)$$

On considère qu'un point est voisin à lui-même. De la même manière,

Définition 2 (Voisinage $\mathcal{N}_2(p)$). *Soit p un point de coordonnées (x_p, y_p) . Le voisinage de type 2 du point p (noté $\mathcal{N}_2(p)$) est l'ensemble des points q de coordonnées (x_q, y_q) qui vérifient :*

$$\max(|x_p - x_q|, |y_p - y_q|) \leq 1 \quad (1.2)$$

$\mathcal{N}_1(p)$ est constitué de p et de l'ensemble des 4 points situés en haut, à gauche, en bas et à droite de p . Pour $\mathcal{N}_2(p)$ il faut ajouter à $\mathcal{N}_1(p)$ les 4 points diagonaux du point p [47, 79]. La figure 1.2 représente ces deux types de voisinage, le point p est représenté par le point central noir.

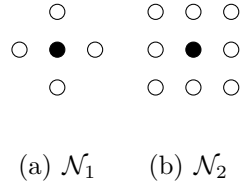


FIGURE 1.2 – Voisinages de type 1 et 2

Grâce aux précédentes définitions, on peut relier 2 points entre eux avec un chemin.

Définition 3 (α -chemin). *Un α -chemin du point p au point q est une suite de points $p = p_0, p_1, p_2, \dots, p_n = q$ telle que p_i est α -voisin (avec $\alpha \in \{4, 8\}$) à p_{i-1} pour $1 \leq i \leq n$.*

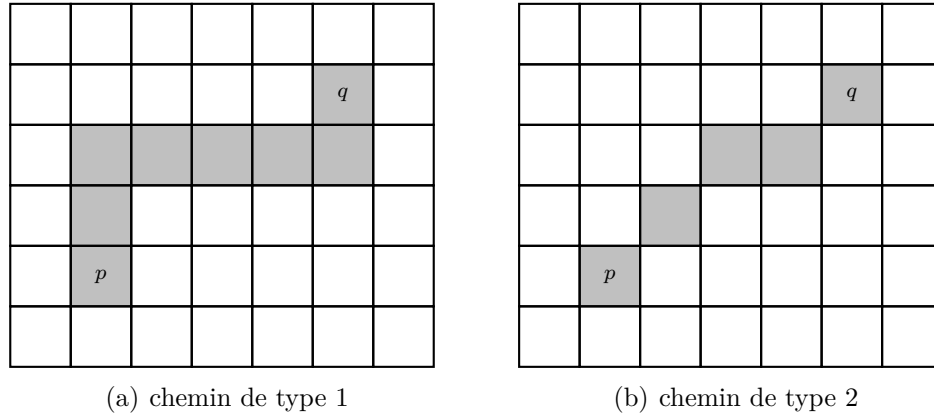


FIGURE 1.3 – Exemples de chemin de type 1 et 2.

1.3 Notion de topologie d'une image

La topologie vient des mots “*topos*” (lieu) et “*logos*” (étude), elle vise à étudier les lieux. Plus précisément, la topologie permet d'émettre une équivalence entre deux objets si l'on peut transformer l'un en l'autre grâce à la rotation, translation, réflexion, changement de taille/volume, etc. Pour que les deux objets restent topologiquement équivalents, la transformation ne doit pas induire de trou, déchirements, sections. La figure 1.4 présente trois objets binaires. Dans cet exemple, la pomme est topologiquement équivalente au marteau. La paire de ciseaux n'est pas topologique à la pomme ni au marteau parce que la paire de ciseaux présente deux trous alors que les deux autres objets n'ont pas de trous.



FIGURE 1.4 – Formes binaires.

Dans cette section sont abordées des notions relatives à la topologie de l'image. Dans un premier temps, dans la sous-section 1.3.1, une définition des composantes

connexes et un algorithme permettant de les identifier sont présentés. En second lieu, dans la sous-section 1.3.2, le nombre d'Euler et son calcul sont présentés. Enfin, dans la sous-section 1.3.3 la notion de point simple est présentée ainsi que différentes méthodes pour le caractériser.

1.3.1 Les composantes connexes

Dans [79], Rosenfeld introduit la notion de composantes connexes (exemple figure 1.5).

Définition 4 (α -connexité). *Soit X un ensemble de points. Deux points sont α -connectés si il existe un α -chemin dans X les reliant.*

Définition 5 (Composante α -connexe). *Une composante α -connexe est un ensemble maximal de points α -connectés dans X .*

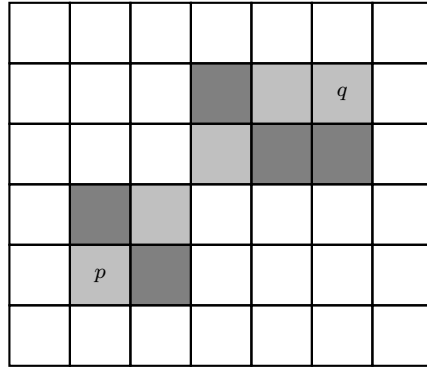


FIGURE 1.5 – Composante 8-connexe.

Dans l'exemple de la figure 1.5 on peut voir que le chemin reliant les points p et q est de type 2. Donc on peut déduire que la forme X est une composante 8-connexe. Comme il n'existe pas de chemin de type 1 reliant ces deux points, X se compose de deux composantes 4-connexes.

Dans sa contribution, Rosenfeld introduit également la “*labelisation*” ou fonction d'étiquetage. Comme son nom l'indique, cette fonction distribue une étiquette unique à chaque composante connexe. Si on étiquette à partir de 1 et par ordre

croissant, cela veut dire que la valeur maximale des étiquettes (ou des labels) correspond au nombre de composantes connexes. La figure 1.6 représente une image composée de 3 composantes connexes étiquetées.

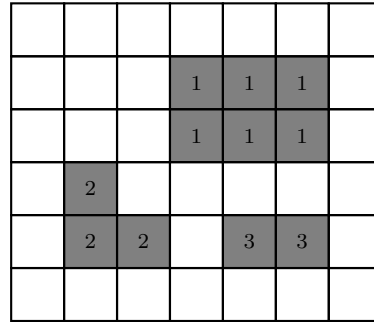


FIGURE 1.6 – Image “*labelisée*”.

Dans sa méthode Rosenfeld parcourt l'image (de gauche à droite et de haut en bas) et lorsqu'il rencontre un point de la forme, il propage l'étiquette de son voisin haut. Si le point en question n'a pas de voisin il crée une nouvelle étiquette et il attribue à ce point cette nouvelle étiquette comme nous le montre la figure 1.7. Dans cette figure les points déjà balayés sont identifiés en gris clair et ont un label contrairement aux autres points.

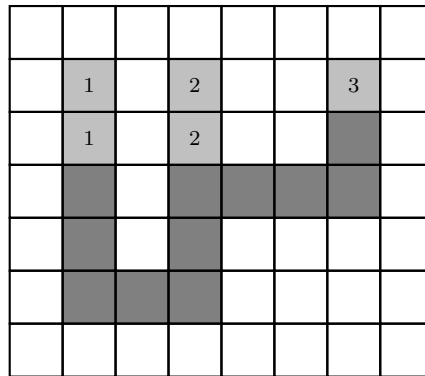


FIGURE 1.7 – Processus de labelisation.

Les étiquettes sont stockées dans un tableau qu'il agrandit à chaque nouvelle étiquette créée. Ainsi lorsque toute l'image a été balayée, il procède à une simplification du tableau d'étiquettes.

1.3.2 Le nombre d'Euler

Le nombre d'Euler, noté E , ou encore la caractéristique d'Euler est un invariant topologique [48].

$$E = S - A + F \quad (1.3)$$

Avec S , A et F qui désignent respectivement le nombre de sommets, d'arêtes et de faces.

Dans une image binaire 2D la propriété d'Euler-Poincaré s'exprime de la façon suivante :

$$E = O - T \quad (1.4)$$

Où O désigne le nombre de composantes connexes dans l'image et T le nombre de trous. Un trou est une composante α -connexe dans le complémentaire de l'image. La figure 1.8 présente deux exemples d'objets binaires. Le nombre d'Euler de la configuration 1.8(a) vaut 0 puisque $E = 2 - 2$ tandis que dans la configuration illustrée dans 1.8(b), $E = 1 - 3$.

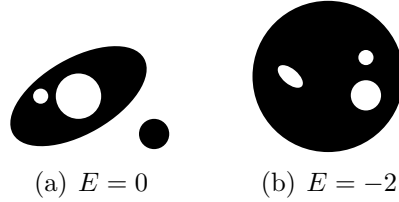


FIGURE 1.8 – Exemple d'objets et leurs nombre d'Euler.

Dans la pratique, un algorithme [41] somme des configurations de pixels pour extraire le nombre d'Euler. Il compte les angles qui forment une convexité et soustrait les angles formants une cavité. Pour cela il utilise des configurations, dans un voisinage 2×2 , pondérées. Ces configurations sont illustrées dans la figure 1.9, il faut également ajouter à ces configurations leurs rotations à 90° , 180° et -90° .

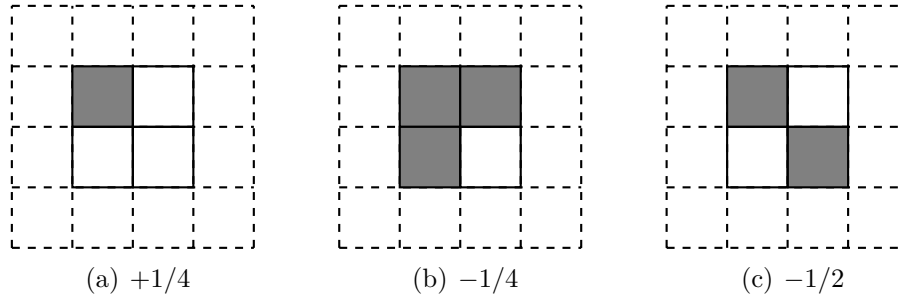


FIGURE 1.9 – Masques 2×2 de configurations pour déterminer le nombre d'Euler.

La figure 1.10 présente deux exemples. Dans le premier, on compte 5 configurations de type 1.9(a) et une configuration de type 1.9(b). Le nombre d'Euler est $E = \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} - \frac{1}{4} = 1$.

Le nombre d'Euler du second exemple est ainsi $E = \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} - \frac{1}{4} - \frac{1}{4} - \frac{1}{4} - \frac{1}{4} = 0$.

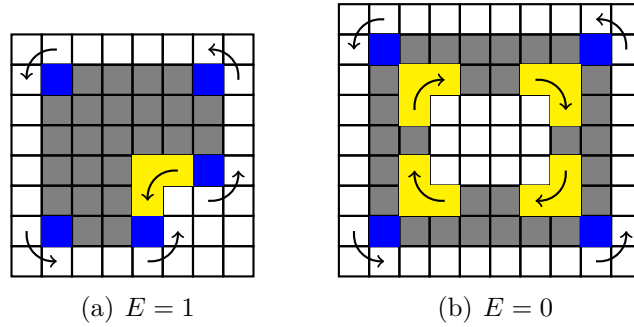


FIGURE 1.10 – Exemples du calcul du nombre d'Euler.

1.3.3 Points simples

Cette section présente ce qu'est un point simple et comment on le caractérise. Cette notion est utilisée dans le processus d'extraction du squelette, introduit dans le chapitre 2.

Définition

La notion de point simple, introduite par Rosenfeld dans [77], permet d'identifier les points pouvant être supprimés sans changer la topologie de la forme.

Définition 6 (Point simple). Soit X une partie de \mathbb{Z}^2 avec un nombre fini de composantes α -connexes et de composantes $\bar{\alpha}$ -connexes. Un point A de X est α -simple dans X si :

- l'ensemble X et $X \setminus \{A\}$ ont le même nombre de composantes α -connexes ;
- l'ensemble \bar{X} et $\bar{X} \setminus \{A\}$ ont le même nombre de composantes $\bar{\alpha}$ -connexes.

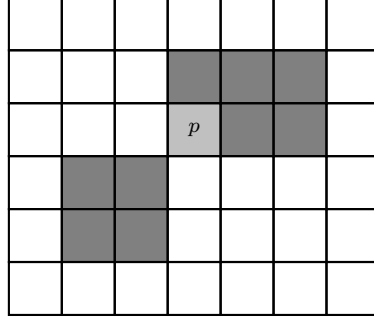


FIGURE 1.11 – Exemple où le point p est 4-simple mais pas 8-simple.

La figure 1.11 illustre un objet et identifie un point p . Si l'on considère la 8-connexité de la forme et la 4-connexité du fond, le point p ne peut pas être supprimé sans scinder la forme en deux. Dans cette configuration, le point p n'est pas 8-simple.

En revanche, si l'on considère la 4-connexité de la forme et la 8-connexité du fond, la forme est composée de deux composantes 4-connexes. La suppression du point p ne change alors pas la topologie de la forme. Dans cette configuration, le point p est 4-simple.

Cette dernière notion est importante dans le cadre de la squelettisation qui demande la préservation de la topologie de la forme (voir le chapitre 2).

Caractérisation des points simples

Il existe plusieurs méthodes pour déterminer si un point est simple. Dans la littérature on appelle “*crossing number*” le nombre de connexité dans le voisinage du point p , permettant de définir la simplicité d'un point. En effet le “*crossing number*” renseigne, grâce au nombre de voisins du point p à étudier, la topologie

locale du sous ensemble de l'objet.

La méthode de Rutovitz [80] consiste à compter le nombre de changement de valeur d'un pixel à l'autre lors du parcours du voisinage de p .

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

FIGURE 1.12 – Représentation du voisinage du point p .

$$X_R(p) = \sum_{i=1}^8 \|x_{i+1} - x_i\| \quad (1.5)$$

Avec $x_9 = x_1$. Le point p est simple si et seulement si $X_R(p) = 2$.

La figure 1.12 montre l'indexation, correspondante à l'équation, des voisins de p .

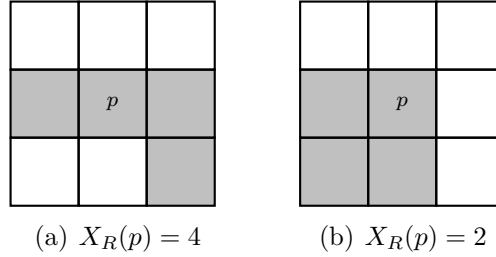


FIGURE 1.13 – Exemple où le crossing number de Rutovitz vaut 4 et 2.

La figure 1.13(a) montre un cas où le point p n'est pas simple et le crossing number vaut 4. En revanche quand le point est simple 1.13(b), le crossing number vaut 2. Une autre manière de calculer la simplicité d'un point a été étudiée par Hilditch [45].

À la différence de la méthode de Rutovitz, Hilditch somme des combinaisons de b_i correspondants aux trois points formant le coin du voisinage.

$$X_H(p) = \sum_{i=1}^4 b_i \quad (1.6)$$

$$\text{où } b_i = \begin{cases} 1 & \text{si } x_{2i-1} = 0 \text{ et } (x_{2i} = 1 \text{ ou } x_{2i+1} = 1) \\ 0 & \text{sinon} \end{cases}$$

Le point p est simple si et seulement si $X_H(p) = 1$.

Le tableau 1.1 montre la décomposition du calcul de la simplicité selon Hilditch des exemples présentés dans la figure 1.14.

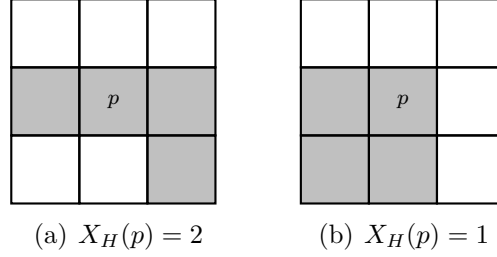


FIGURE 1.14 – Exemple où le crossing number d'Hilditch vaut 2 et 1.

Dans le premier cas (figure 1.14(a)), le crossing number vaut 2 car le point p n'est pas simple. Dans le second cas (figure 1.14(b)), le point p est simple car le crossing number vaut 1.

TABLE 1.1 – Décomposition du calcul de X_H des exemples de la figure 1.14.

	Exemple 1.14(a)				Exemple 1.14(b)			
i	x_{2i-1}	x_{2i}	x_{2i+1}	bi	x_{2i-1}	x_{2i}	x_{2i+1}	bi
1	1	0	0	0	0	0	0	0
2	0	0	1	1	0	0	1	1
3	1	0	0	0	1	1	1	0
4	0	1	1	1	1	0	0	0
$\sum_{i=1}^4 bi$	2				1			

Une autre façon de déterminer si un point est simple ou non est d'utiliser le nombre de Yokoi [98].

Pour le 4-voisinage de la forme et le 8-voisinage du fond le nombre de Yokoi se note Y_4 et pour le 8-voisinage de la forme et le 4-voisinage du fond le nombre de Yokoi se note Y_8 . Les equations 1.7 et 1.8 utilisent l'identification des pixels de la figure 1.12.

$$Y_4 = x_1(1 - x_8x_7) + x_7(1 - x_6x_5) + x_5(1 - x_4x_3) + x_3(1 - x_2x_1) \quad (1.7)$$

$$Y_8 = x'_1(1 - x'_8x'_7) + x'_7(1 - x'_6x'_5) + x'_5(1 - x'_4x'_3) + x'_3(1 - x'_2x'_1) \quad (1.8)$$

où $x'_i = 1 - x_i$.

Avec cette méthode le pixel p est simple si et seulement si le nombre de Yokoi vaut 1.

1.4 Les distances discrètes

Comme dans l'espace continu, il est possible d'effectuer des mesures dans l'espace discret. La distance euclidienne est une mesure de distance entre deux objets A et B dans le domaine continu. Dans le domaine discret, de nombreux travaux ont été réalisés pour définir des distances applicables au domaine discret : les distances discrètes.

Certains travaux s'appliquent à calculer la distance euclidienne d_E dans le domaine discret [35, 72]. Soit deux points p et q et de coordonnées $p = (p_1, \dots, p_n)$ et $q = (q_1, \dots, q_n)$ dans \mathbb{R}^n :

$$d_E(p, q) = \sqrt{(q_1 - p_1)^2 + \dots + (q_n - p_n)^2} \quad (1.9)$$

Dans la suite de ce travail, on se placera dans un contexte séquentiel et en 2 dimensions. C'est pourquoi nous n'aborderons pas les algorithmes de calcul de distance en n -dimensions [23, 37, 72, 76, 86, 88, 89] et parallèles [27, 35, 72, 78].

1.4.1 Définition

Les distances discrètes ont été introduites par Rosenfeld et Pfaltz dans [79]. On définit généralement la distance comme la longueur du plus petit chemin qui relie un point à un autre.

Définition 7 (Distance discrète). *Soit une fonction $d : \mathbb{Z}^n \times \mathbb{Z}^n \rightarrow \mathbb{N}$ et $\forall x, y, z \in \mathbb{Z}^n$. On considère les propriétés suivantes :*

1. **définie positive** $d(x, y) \geq 0$ et $d(x, y) = 0 \Leftrightarrow x = y$,
2. **symétrique** $d(x, y) = d(y, x)$,
3. **inégalité triangulaire** $d(x, z) \leq d(x, y) + d(y, z)$,
4. **invariante en translation** $d(x + z, y + z) = d(x, y)$,

d est appelée une distance si les conditions 1 à 3 sont vérifiées et une distance invariante en translation si toutes les conditions sont vérifiées.

La plupart des distances discrètes peuvent s'exprimer comme la longueur du plus court chemin allant d'un point p à un autre q . Rappelons qu'un chemin est une séquence de points $(p_0 = p, p_1, \dots, p_n = q)$ dont deux points successifs sont adjacents.

Le coût ou la longueur d'un chemin est le nombre de ces déplacements, ici n .

Pour la compréhension de ce chapitre, nous introduisons la notion de boule de distance ou disque (définition 8).

Définition 8 (Disque (ou boule)). *Soient d une distance, p un point dans X et r un rayon. Le disque fermé D_{\leq} est l'ensemble :*

$$D_{\leq}(p, r) = \{q : d(p, q) \leq r\} \quad (1.10)$$

C'est-à-dire que le point q est un point du disque si sa distance à p (le centre de ce disque) est inférieure ou égale à r (son rayon).

On définit également le disque ouvert $D_{<}$:

$$D_{<}(p, r) = \{q : d(p, q) < r\} \quad (1.11)$$

Lorsqu'il n'y a pas de précision sur la dénomination du disque D , on considère un disque fermé. Pour simplifier la notation, le disque $D(r)$ désigne le disque fermé de centre O et de rayon r .

1.4.2 Distances simples

Les premières distances utilisées sont les distances d_4 aussi appelée City Block distance, Manhattan distance ou Square distance et d_8 appelée aussi Chessboard distance ou diamond distance. Dans [79], Rosenfeld les nomme respectivement d et d^* . Dans la suite de ce travail nous les appellerons d_4 et d_8 .

$$d_4(x, y) = \sum_{i=1}^2 |y_i - x_i| \quad (1.12)$$

La distance d_4 est la longueur du plus petit chemin 4-connexe entre deux points : $p = p_0, p_1, p_2, \dots, p_n = q$ où p_n et p_{n-1} appartiennent à \mathcal{N}_1 . La suite de disques associée à cette distance est donnée dans la figure 1.15.

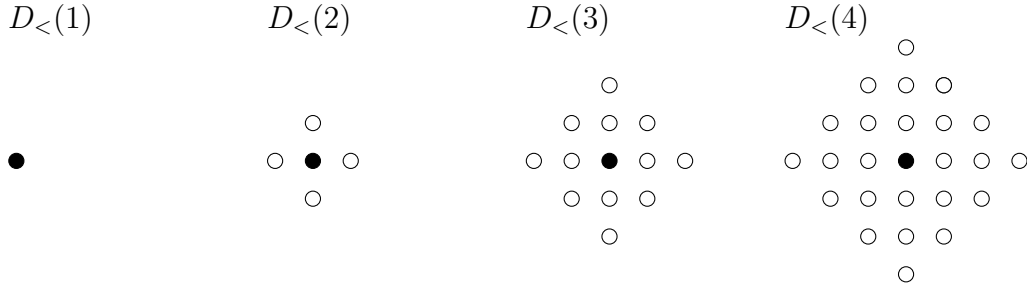
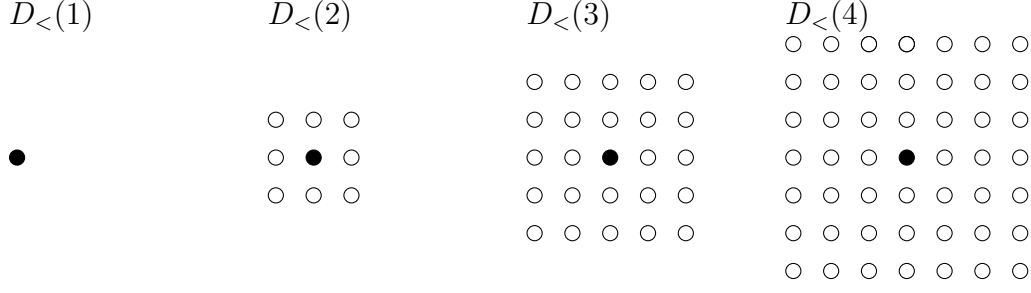


FIGURE 1.15 – Suite de disques d_4 .

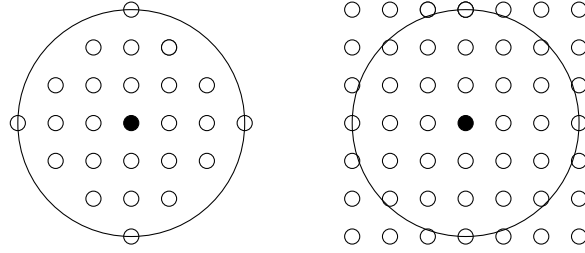
De la même manière que pour d_4 , la distance d_8 est la longueur du plus petit chemin 8-connexe entre deux points : $p = p_0, p_1, p_2, \dots, p_n = q$ où p_n et p_{n-1} appartiennent à $\mathcal{N}_2(p)$.

$$d_8(x, y) = \max_{i=1,2} |y_i - x_i| \quad (1.13)$$

Cette distance permet des déplacements verticaux, horizontaux et diagonaux. La suite de disques correspondant à cette distance est représentée 1.16


FIGURE 1.16 – Suite de disques d_8 .

Mais ces distances sont une mauvaise estimation de la distance euclidienne car d_4 sur-estime cette distance en diagonale alors que d_8 la sous-estime par rapport aux chemins verticaux et horizontaux. La figure 1.17 montre un disque d_4 et un autre disque d_8 et une boule euclidienne de même rayon. Ainsi on peut se rendre compte de la différence des distances discrètes simples par rapport à une distance euclidienne.


FIGURE 1.17 – Disques d_4 , d_8 et boule euclidienne.

Pour remédier à ce problème, il existe deux autres types de distances. La première que nous allons étudier est la distance à séquence de voisinage.

1.4.3 Distances à séquence de voisinage

Les distances à séquence de voisinage, notées d_{ns} , utilisent les deux types de voisinage \mathcal{N}_1 et \mathcal{N}_2 . La séquence utilisée, notée B par la suite, est une combinaison de \mathcal{N}_1 et de \mathcal{N}_2 .

Cette séquence est utilisée pour créer une suite de disques en faisant une somme de Minkowski avec le dernier disque calculé de la suite et l'un des deux voisinages (\mathcal{N}_1 ou \mathcal{N}_2) selon la séquence B .

La somme de Minkowski 1.14 associée à deux parties P_1 et P_2 l'ensemble formé des sommes d'éléments de P_1 et de P_2 .

$$P_1 \oplus P_2 = \{p_1 + p_2 \mid p_1 \in P_1, p_2 \in P_2\} \quad (1.14)$$

L'équation 1.15 permet de calculer la suite de disques.

$$D(p, 0) = \{p\}, D(p, r) = D(p, r - 1) \oplus \mathcal{N}_{B(r)} \quad (1.15)$$

Le premier disque de la séquence contient un seul point, le centre de ce disque p . Pour calculer les autres disques il suffit d'utiliser le disque de rayon inférieur et de faire une somme de Minkowski avec le voisinage correspondant au rayon du disque que l'on cherche à calculer. La figure 1.18 décrit le calcul de $D_{<}(3)$. P_1 représente le disque $D_{<}(2)$ dans \mathcal{N}_1 et P_2 représente le disque $D_{<}(2)$ dans \mathcal{N}_2 .

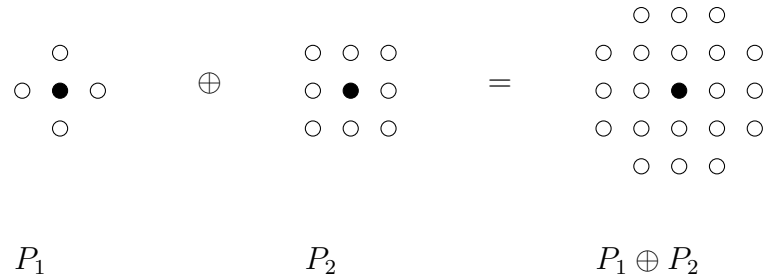


FIGURE 1.18 – Exemple $P_1 \oplus P_2$.

La figure 1.19 illustre les disques utilisés pour créer les chemins suivant la distance utilisée :

- Pour la distance d_4 , le voisinage utilisé pour faire la somme de Minkowski est \mathcal{N}_1 .
- Pour la distance d_8 le voisinage utilisé pour faire la somme de Minkowski est \mathcal{N}_2 .
- Pour une distance d_{NS} le voisinage utilisé pour faire la somme de Minkowski

est défini par la séquence B . Dans cet exemple il s'agit de la distance octogonale (d_{oct}). Dans cet exemple, $B = (\overline{1,2})$ donc les voisinages utilisés pour faire la somme de Minkowski sont \mathcal{N}_1 et \mathcal{N}_2 alternativement.

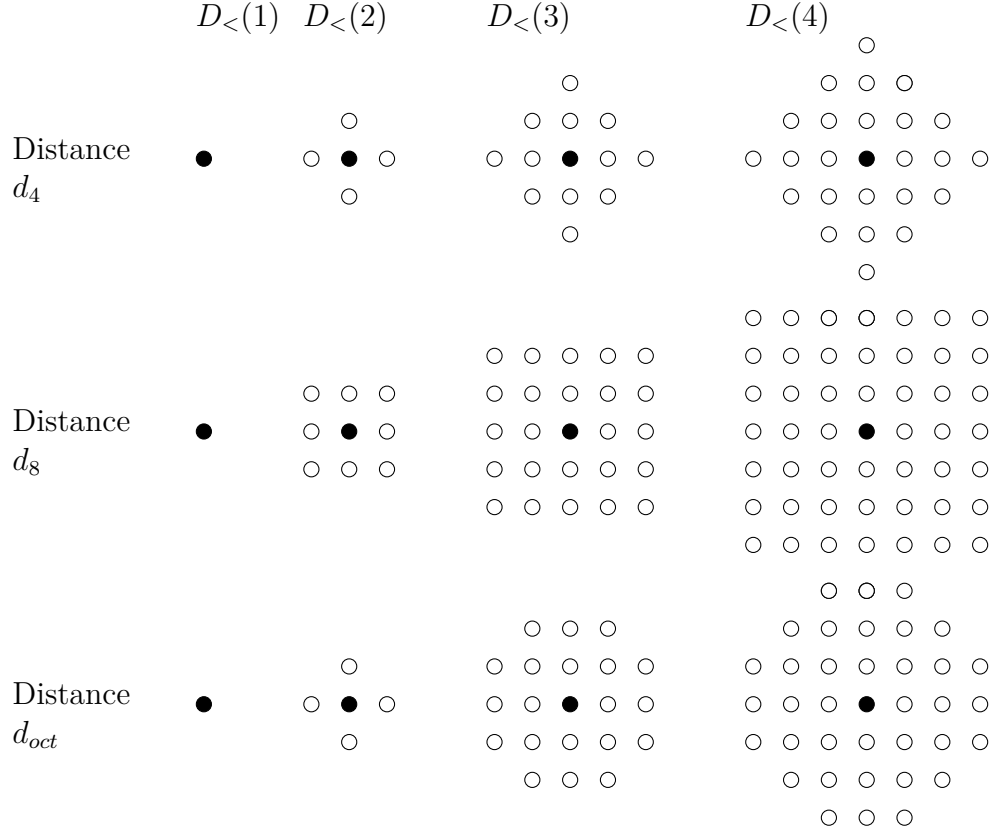


FIGURE 1.19 – Suites de disques de rayons 1 à 4 correspondantes aux distances d_4 , d_8 et d_{oct} .

Dans son travail de thèse, Nicolas Normand [57] calcule une distance à séquence de voisinage grâce à une succession d'opérations de Minkowski. La particularité de son algorithme réside dans le fait qu'il décentre le centre des disques. Ainsi il peut calculer une distance selon un seul balayage de l'image.

Rosenfeld introduit la distance octogonale, dans [78], qui utilise une séquence de voisinage alternant successivement les voisinages \mathcal{N}_1 et \mathcal{N}_2 . On notera cette séquence $B = (\overline{1,2})$. Cette distance a été étudiée plus précisément par la suite par

Das dans [36] et ses travaux ont été généralisés à n dimensions par Nagy dans [56].

Wang et Bertrand [97] calculent d_{oct} grâce à une succession d'opérations morphologiques (Minkowski) utilisant alternativement différents éléments structurants.

Contrairement aux autres approches, Hajdu et Hajdu proposent des distances à séquence de voisinages non périodiques dans [44]. Pour ce faire, ils utilisent les séquences de Beatty pour obtenir une séquence et ainsi créer une suite de disques.

La séquence de Beatty [16, 17] de paramètre irrationnel α est la séquence des parties entières des multiples de α :

$$\lfloor \alpha \rfloor, \lfloor 2\alpha \rfloor, \lfloor 3\alpha \rfloor, \dots = (\lfloor n\alpha \rfloor)_{n \geq 1} \quad (1.16)$$

Ils définissent une séquence de voisinage comme la séquence des écarts de la suite de Beatty :

$$B(r) = \lfloor r\alpha \rfloor - \lfloor (r-1)\alpha \rfloor \quad (1.17)$$

avec $1 \leq \alpha \leq 2$.

On utilise la fonction $\mathbf{1}_B$ comme compteur d'apparitions de la valeur 1 dans la séquence B (équation 1.18). De la même manière $\mathbf{2}_B$ compte le nombre d'apparitions de la valeur 2 dans la séquence (équation 1.19). Donc pour tous les r :

$$\mathbf{1}_B(r) = \text{card}\{i : B(i) = 1, 1 \leq i \leq r\} \quad (1.18)$$

et

$$\mathbf{2}_B(r) = \text{card}\{i : B(i) = 2, 1 \leq i \leq r\} \quad (1.19)$$

Par commodité on écrit $\mathbf{1}_B(0) = \mathbf{2}_B(0) = 0$. On notera que $\mathbf{1}_B(r) + \mathbf{2}_B(r) = r$ avec $r \in \mathbb{N}$. Le tableau 1.2 montre une séquence $B(r)$ et ses correspondants $\mathbf{1}_B$ et $\mathbf{2}_B$.

TABLE 1.2 – Suite de Beatty

r	1	2	3	4	5	6	7	8
B	1	2	1	2	1	2	1	2
$\mathbf{1}_B$	1	1	2	2	3	3	4	4
$\mathbf{2}_B$	0	1	1	2	2	3	3	4

Pour tout $r \in \mathbb{N}$ on a $\mathbf{2}_B(r) = \lfloor (\alpha - 1)r \rfloor$.

On notera que l'utilisation de $\alpha = 3/2$ produit la séquence $B = (\overline{1}, \overline{2})$.

$$B(r) = \lfloor 3/2r \rfloor - \lfloor 3/2(r-1) \rfloor = \lfloor r/2 \rfloor - \lfloor (r-1)/2 \rfloor + 1$$

La figure 1.20 présente les disques de la distance octogonale.

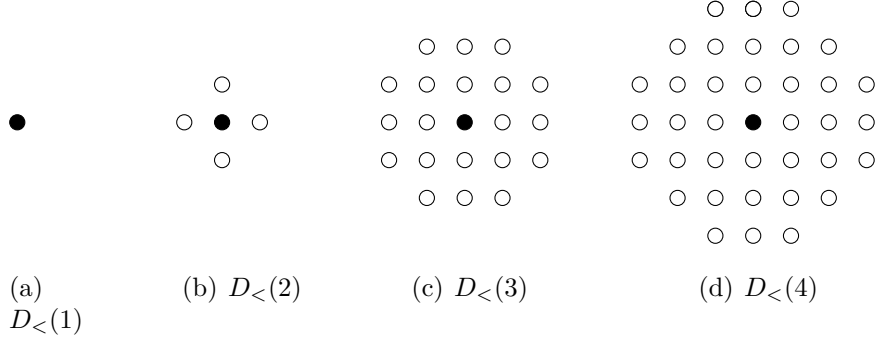


FIGURE 1.20 – Suite de disques d_{NS} avec $B = (\overline{1}, \overline{2})$.

1.4.4 Distances de chanfrein

La famille des distances de chanfrein a été introduite par Montanari dans [54] qui propose une distance qui utilise des masques pondérés. On les appelle également les distances pondérées. Le principe de cette distance est de donner des poids aux voisinages. La figure 1.21 illustre différents masques de chanfrein, noté \mathcal{M} , avec leurs poids $w_i = a, b$ et c .

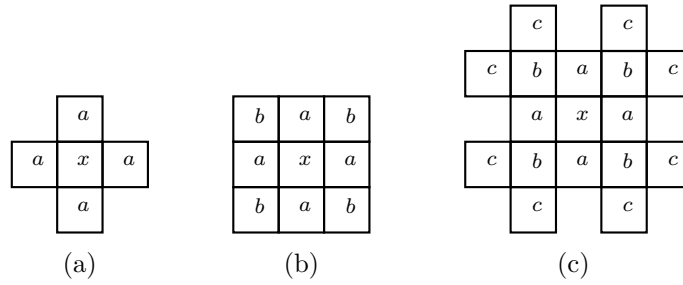


FIGURE 1.21 – Masques de chanfrein.

Borgefors [22] popularise cette distance grâce à des poids spécifiques comme $d_{3,4}$ (où $a = 3$ et $b = 4$) ou encore $d_{5,7,11}$ (où $a = 5$, $b = 7$ et $c = 11$).

Pour calculer la carte de distance de chanfrein de manière séquentielle il faut décomposer les masques. Les figures 1.22 et 1.23 montrent respectivement la décomposition du masque \mathcal{M} en un demi masque avant \mathcal{M}_{av} puis en demi masque arrière \mathcal{M}_{ar} .

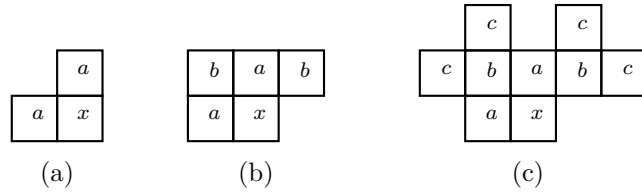


FIGURE 1.22 – Masque avant \mathcal{M}_{av} .

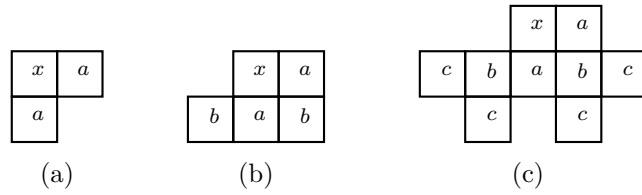


FIGURE 1.23 – Masque arrière \mathcal{M}_{ar} .

L'algorithme 1 nous indique comment calculer séquentiellement la carte dis-

tance grâce aux demi masques de chanfrein.

Algorithme 1: Calcul de la carte de distance de chanfrein

Données : Image binaire X

Résultat : Carte de distance DT_X

```

1 % Balayage direct
2 pour  $l \leftarrow 0$  à  $L - 1$  faire
3   | pour  $k \leftarrow 0$  à  $K - 1$  faire
4   |   | si  $(l,k) \in X$  alors
5   |   |   |  $DT(l,k) \leftarrow \min_{((x,y),w) \in \mathcal{M}_{av}} \{DT(l+x, k+y) + w\};$ 
6   |   |   fin
7   |   fin
8   fin
9 % Balayage rétrograde
10 pour  $l \leftarrow L - 1$  à  $0$  faire
11   | pour  $k \leftarrow K - 1$  à  $0$  faire
12   |   | si  $(l,k) \in X$  alors
13   |   |   |  $DT(l,k) \leftarrow \min\{DT(l, k), \min_{((x,y),w) \in \mathcal{M}_{ar}} DT(l+x, k+y) + w\};$ 
14   |   |   fin
15   |   fin
16 fin

```

Nous ne détaillerons pas plus cette distance dans ce travail. Cependant pour plus d'informations la concernant, les lecteurs intéressés pourront lire la thèse d'Édouard Thiel [94] et son mémoire d'Habilitation à Diriger des Recherches [93] ou encore la thèse d'Éric Rémy [75] où la distance est calculée sur un volume.

1.5 Transformation en distances

À partir des différentes distances, il est possible de créer une carte de distance ou image de distance. Dans la suite de ce travail nous noterons la carte de distance d'une image X , DT_X . La transformation en distance consiste à attribuer à chaque pixel le rayon de la plus grande boule ouverte (équation 1.11) contenue dans la forme dont le centre est ce pixel. En d'autres termes, on affecte à chaque pixel sa distance au point du fond le plus près.

La figure 1.24 illustre un exemple de carte de distance.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	2	2	2	2	1	0	0
0	0	1	2	2	2	3	3	2	2	1	0
0	1	2	2	2	2	3	4	3	2	1	0
0	1	2	1	1	2	2	3	2	2	1	0
0	1	1	0	0	1	2	2	2	1	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 1.24 – Exemple de carte de distance (d_{oct}).

Dans la pratique, deux balayages de l'image sont nécessaires pour calculer une carte de distance simple (d_4 , d_8). Le premier, dit "balayage direct", propage la distance avec la moitié haute du disque en balayant de gauche à droite et de haut en bas de l'image. Le deuxième balayage, dit "balayage rétrograde" réajuste les valeurs de la carte de distance avec la moitié basse du disque en balayant l'image de droite à gauche et de bas en haut. La figure 1.25 illustre les demi-disques hauts et bas pour des disques de type 1 et 2.

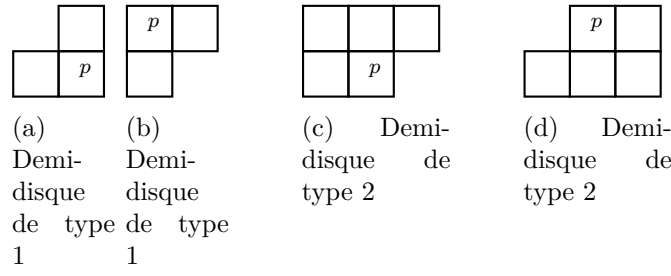


FIGURE 1.25 – (a) et (b) utilisés lors du balayage direct, (c) et (d) pour le balayage rétrograde.

La figure 1.26 représente l'état de la carte de distance après le balayage direct. Cet exemple de carte a été calculé avec le demi disque 1.25(a).

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	2	2	2	2	1	0	0	0
0	0	1	2	2	3	3	3	3	2	1	0	0
0	1	2	3	3	4	4	4	4	3	2	0	0
0	1	2	3	4	5	5	5	5	4	3	0	0
0	1	2	0	0	1	2	3	4	5	0	0	0
0	0	0	0	0	0	1	2	3	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 1.26 – Exemple de carte de distance après le 1^{er} balayage.

On note que les valeurs de la carte sont fausses à mesure où on la lit de gauche à droite et de bas en haut. Par exemple le dernier pixel de la forme a une valeur $DT_X = 3$, or la distance de ce pixel au fond est 1. Il suffit donc d'utiliser le disque 1.25(b) et de procéder au balayage rétrograde pour réajuster les valeurs de la carte comme le montre la figure 1.27.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	2	2	2	2	1	0	0
0	0	1	2	2	3	3	3	3	2	1	0
0	1	2	2	2	3	4	4	3	2	1	0
0	1	2	1	1	2	3	3	3	2	1	0
0	1	1	0	0	1	2	2	2	1	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 1.27 – Exemple de carte de distance après le 2nd balayage.

L'algorithme 2 montre les différentes itérations pour calculer la carte de distance d_4 . De la même manière, pour calculer la carte de distance d_8 il suffit de changer les demis disques.

D'autre part, le calcul d'une carte de distance à séquence de voisinage se calcule différemment. D'après [35] il faut quatre balayages de l'image pour obtenir une telle carte.

Afin de simplifier l'algorithme nous allons identifier les points du voisinage. La figure 1.28 illustre cette identification.

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

FIGURE 1.28 – Identification des points de voisinage du point p .

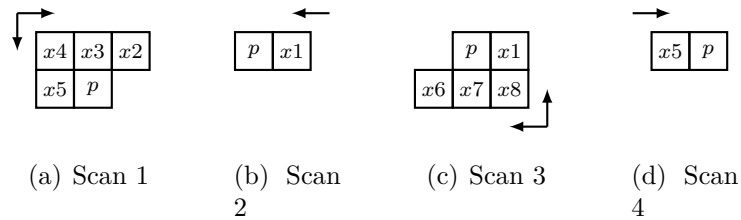


FIGURE 1.29 – Sens de balayages de l'image de l'algorithme de référence.

Algorithme 2: Calcul de la carte de distance d_4

Données : Image binaire X **Résultat :** Carte de distance DT_X

```
1 % Balayage direct
2 pour  $l \leftarrow 0$  à  $L - 1$  faire
3   | pour  $k \leftarrow 0$  à  $K - 1$  faire
4   |   | si  $(l,k) \in X$  alors
5   |   |   |  $DT_X(l,k) \leftarrow \min(DT_X(l-1,k)+1, DT_X(l,k-1)+1);$ 
6   |   |   fin
7   |   fin
8   fin
9 % Balayage rétrograde
10 pour  $l \leftarrow L - 1$  à  $0$  faire
11   | pour  $k \leftarrow K - 1$  à  $0$  faire
12   |   |  $DT_X(l,k) \leftarrow \min(DT_X(l,k), DT_X(l+1,k) + 1, DT_X(l,k+1) + 1);$ 
13   |   fin
14 fin
```

Algorithme 3: Calcul de la carte de distance d_{oct}

Données : Image binaire X

Résultat : Carte de distance DT_X

```

1 pour  $p$  dans l'ordre scan1 faire
2    $DT_X(p) \leftarrow 0$  si  $p \in X(p)$ 
3    $DT_X(p) \leftarrow 255$  si  $p \notin X(p)$ 
4 pour  $p$  dans l'ordre scan1 faire
5    $DT_X(p) = \min \begin{cases} DT_X(p) \\ DT_X(x_5) + 1 \\ DT_X(x_3) + 1 \\ DT_X(x_4) + 1 \text{ si } DT_X(x_4) \text{ est paire} \\ DT_X(x_2) + 1 \text{ si } DT_X(x_2) \text{ est paire} \end{cases}$ 
6   pour  $p$  dans l'ordre scan2 faire
7      $DT_X(p) = \min \begin{cases} DT_X(p) \\ DT_X(x_1) + 1 \end{cases}$ 
8 pour  $p$  dans l'ordre scan3 faire
9    $DT_X(p) = \min \begin{cases} DT_X(p) \\ DT_X(x_1) + 1 \\ DT_X(x_7) + 1 \\ DT_X(x_8) + 1 \text{ si } DT_X(x_8) \text{ est paire} \\ DT_X(x_6) + 1 \text{ si } DT_X(x_6) \text{ est paire} \end{cases}$ 
10  pour  $p$  dans l'ordre scan4 faire
11     $DT_X(p) = \min \begin{cases} DT_X(p) \\ DT_X(x_5) + 1 \end{cases}$ 

```

1.6 Axe médian

L'axe médian est l'ensemble des centres des boules maximales incluses dans la forme. L'extraction de l'axe médian est une opération réversible si l'on garde

l'information de distance en chaque point. C'est-à-dire que l'on peut obtenir la forme originale grâce à l'axe médian, ce qui fait de lui un outil de compression d'image.

La notion d'axe médian a été introduite par Blum dans [21] avec l'analogie du feu de prairie.

Définition 9 (Analogie du feu de prairie). *Soit une prairie couverte de manière homogène par de l'herbe sèche et p un ensemble de points de cette prairie. Au départ, tous les points du contour de p sont enflammés simultanément. Le feu se propage de manière homogène et s'étend à travers la prairie à une vitesse constante. le squelette de l'ensemble de points p est défini comme le lieu des points où les fronts enflammés se sont rencontrés*

Rosenfeld [79] définit l'axe médian comme l'ensemble des maxima locaux de la carte de distance.

Définition 10 (Axe médian). *L'axe médian d'une image binaire X est l'ensemble des centres des boules ouvertes maximales dans X pondérées par le rayon r .*

On peut le représenter par la fonction ci-dessous notée MA_X :

$$\begin{aligned} \text{MA}_X : \mathbb{Z}^n &\rightarrow \mathbb{N} \\ \text{MA}_X(p) &= \begin{cases} 0 & \text{si } \exists q, r : D_{<}(p, \text{DT}_X(p)) \subsetneq D_{<}(q, r) \subset X, \\ \text{DT}_X(p) & \text{sinon.} \end{cases} \end{aligned} \quad (1.20)$$

Cependant il n'est pas suffisant d'identifier les maxima locaux de la carte de distance pour trouver les points de l'axe médian. Cela dépend de la distance utilisée pour calculer DT_X . Par exemple, l'utilisation des distances de chanfrein nécessite plusieurs opérations afin d'extraire l'axe médian [94].

La figure 1.30 présente un exemple de l'axe médian (points encerclés) calculé à partir d'une carte de distance d_4 .

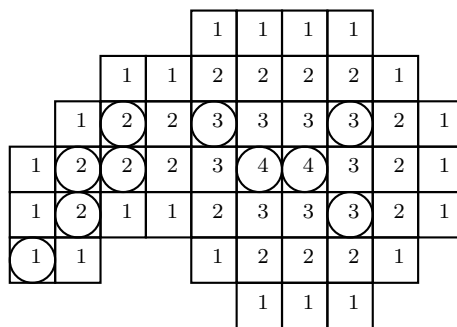


FIGURE 1.30 – Exemple de point de l'axe médian sur une carte de distance d_4 .

Un disque maximal peut être couvert par l'union de plusieurs disques maximaux mais il existe des algorithmes permettant d'affiner cet axe médian[55, 90].

Dans leur travail, Svensson, Borgefors et Nyström [90] nomment cet axe médian le CMD pour “center of a maximal disks”. Elles calculent ensuite un ensemble de CMD réduit (RCMDs). Pour cela, elles listent les CMDs, les trient en fonction de leur distance par ordre croissant et sauvegardent leur position. Elles créent une image temporaire où elles enregistrent le nombre de disques couvrant chaque pixel de l'image originale. Enfin elle parcourent la liste des CMDs (par ordre croissant de distance) et inspectent l'image temporaire pour vérifier si le disque correspondant couvre les pixels qui ont une valeur supérieure à 1. Si c'est le cas, elles suppriment le CMD de la liste. La figure 1.31 présente un exemple où les CMDs sont représentés en gris et les RCMDs sont encadrés.

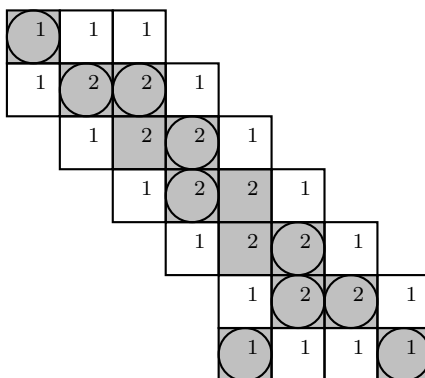


FIGURE 1.31 – Exemple de CMDs et RCMDs.

Il a été démontré, par une réduction polynomiale du problème SAT-3, que le

calcul d'un axe médian minimal était un problème NP-difficile [31].

1.7 Propagation de l'information

Dans cette section, nous allons aborder la notion de propagation de l'information selon différents angles de vue.

Tout d'abord, on vient de voir dans la section 3.3 le calcul d'une carte de distance. Si deux balayages de l'image sont nécessaires pour calculer la carte c'est à cause de la propagation de l'information dans l'image.

D'autre part, dans cette section, la notion de dépendance entre pixels est décrite. Lors de la caractérisation des points simples, l'ordre de balayage de l'image joue un rôle très important dans le résultat.

Lors du calcul de la carte de distance (pour une distance simple comme d_4 ou d_8), deux balayages de l'image sont nécessaires. Le balayage direct utilise la moitié haute du disque et le balayage rétrograde utilise la moitié basse de ce même disque. Le résultat intermédiaire de ce processus est la "carte de distance" produite après le balayage direct, juste avant le balayage rétrograde. Un exemple de résultat intermédiaire est donné en figure 1.26. Nous nous sommes rendu compte que les valeurs de distance étaient croissantes à mesure où on avançait dans l'image dans l'ordre "raster scan". Cependant, si on avait inversé les balayages, c'est à dire que si on avait commencé à balayer de bas en haut et de droite à gauche dans le premier balayage puis de haut en bas et de gauche à droite dans le second, nous n'aurions pas obtenu le même résultat intermédiaire (figure 1.32) mais le même résultat final.

Dans ce résultat intermédiaire on remarque que les valeurs de distance sont croissantes à mesure où on lit dans l'ordre "raster scan" inversé.

Cette expérience montre que la propagation de l'information dépend fortement de l'ordre du balayage de l'image.

D'une autre part, la dépendance entre pixels lors de la caractérisation des points simples dans le cadre d'une méthode de Thinning (voir section 2.3) a été étudiée [74]. L'ordre de balayage de l'image joue un rôle important dans le processus de caractérisation des points simples.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	4	3	2	1	0	0	0
0	0	0	4	4	5	4	3	2	1	0	0
0	0	4	3	3	4	5	4	3	2	1	0
0	3	3	2	2	3	4	4	3	2	1	0
0	2	2	1	1	2	3	3	3	2	1	0
0	1	1	0	0	1	2	2	2	1	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 1.32 – Exemple de résultat intermédiaire si les balayages étaient inversés.

Si on admet une figure 1.33 contenant 2 points. Les points p et q sont simples. Dans l'ordre de balayage de l'image (gauche à droite et de haut en bas), le premier point rencontré est p . Il est donc supprimé et sa suppression change la simplicité du point q . Dans les mêmes conditions, si cette image subit une rotation de 180 degrés alors le point q (visitée en premier) se retrouve supprimé. Ceci engendre alors la non simplicité du point p .

En somme, la suppression d'un pixel entraîne une modification de l'état de son voisin.

0	0	0	0	0	0	0	0
0	p	q	0	0	q	p	0
0	0	0	0	0	0	0	0

(a) image contenant 2 points

(b) image après rotation de 180

FIGURE 1.33 – Exemple où le point p est simple dans un ordre de balayage mais pas dans un autre.

Par cet exemple on s'aperçoit que ce cas se présente également si on admet la première configuration de 1.33 mais avec deux ordres de balayages différents. Avec un raster scan standard, p est simple et q ne l'est pas. Et de la même façon, avec

un raster scan inversé (balayage de bas en haut et de droite à gauche), q est simple et p ne l'est pas.

Cette notion de dépendance entre pixel intiment liée à l'ordre de balayage est importante pour la compréhension de la suite de ce travail.

1.8 Conclusion

Ce chapitre a introduit l'espace dans lequel nous allons travailler : l'espace discret. De cet espace découlent de nouvelles définitions discrètes comme la notion de voisinage de points qui conduit aux chemins puis aux distances discrètes.

Depuis les années 60 de nombreux chercheurs ont visé à décrire un espace de travail discret en définissant des relations entre des points de ce plan. La connexité d'un point est une clé maîtresse de tout travail en morphologie mathématique. Elle décrit comment un point est collé à un autre par 2 types de voisinages \mathcal{N}_1 ou \mathcal{N}_2 . Cette définition de voisinage est utile pour décrire une forme, définir la simplicité d'un point, définir un chemin et ainsi une distance.

Ces notions de base sont, dans ce travail, utilisées pour mettre en pratique un célèbre descripteur de forme : le squelette.

La squelettisation

2.1 Introduction

L'espace de travail de cette thèse étant introduit, nous allons nous intéresser à l'élément central de ce travail : la squelettisation.

La squelettisation est le nom apporté à la méthode qui consiste à extraire le squelette d'une forme binaire.

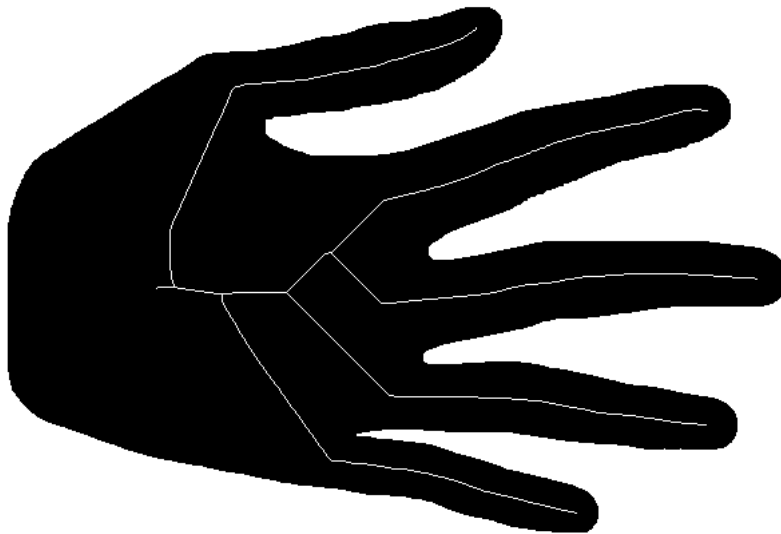


FIGURE 2.1 – Squelette d'une forme de main.

Le squelette (définition 11) est une représentation compacte et efficace de la forme. Il est largement utilisé dans le domaine de l'analyse d'image car c'est un

puissant descripteur de forme. De plus, si l'information de distance est préservée, l'opération de squelettisation est réversible.

Le squelette est aussi utilisé en compression d'image. L'axe médian étant lui aussi un outil performant pour la compression de l'image, il ne décrit pas la forme car il ne conserve pas la topologie de celle-ci.

Dans le cas 2D, le squelette représente la ligne centrée dans la forme comme le montre la figure 2.1.

Après une définition du squelette et de ses propriétés dans la section 2.2, ce chapitre présentera 3 types de méthodes séquentielles de squelettisation 2D.

La première méthode est l'amincissement et est présentée dans la section 2.3. La seconde méthode est basée sur les cartes de distance et présentée dans la section 2.4. Enfin, une méthode hybride est présentée dans la section 2.5.

2.2 Le squelette et ses propriétés

La notion de squelette a été introduite pour la première fois par Blum dans [21] avec l'analogie des feux de prairies (voir définition 9).

Définition 11 (Algorithme de squelettisation). *L'algorithme de squelettisation est l'opération qui identifie le squelette d'une forme. Le squelette est un sous-ensemble de la forme qui a les propriétés suivantes :*

- *reconstructible,*
- *d'épaisseur unitaire,*
- *ayant la même topologie que celle-ci,*
- *centré dans la figure.*

Cependant, ces quatre propriétés sont discutables. Il est très difficile d'obtenir les quatre propriétés à la fois, voir même impossible, dans un squelette. Si l'on considère une forme d'épaisseur paire, le squelette sera :

- Soit d'une épaisseur de deux pixels,
- Soit décentré.

Il faut alors émettre un ordre de préférence sur les propriétés du squelette que l'on veut obtenir. Par exemple, plus tard dans ce travail, nous privilégierons le centrage

dans la forme au détriment de la finesse.

Une méthode de squelettisation est ensuite choisie selon les propriétés du squelette exigées. Il existe plusieurs méthodes pour calculer le squelette.

Dans ce travail nous nous plaçons dans un contexte discret, séquentiel avec des images en 2D. Nous n'allons donc pas aborder, dans ce travail, les autres méthodes. Cependant il faut noter que de nombreux travaux ont été publiés :

- Squelettisation dans le domaine continu utilisant les diagrammes de Voronoï [13, 62],
- Squelettisation 3D [25, 28, 40, 64],
- Algorithmes de squelettisation parallèle 2D [33, 99] et 3D [6, 40, 52, 63].

Dans son mémoire de thèse, Christophe Lohou [51] présente et compare différents algorithmes de squelettisation parallèle en 2D et en 3D.

2.3 Squelettisation basée sur l'amincissement (**Thinning**)

L'amincissement ou *thinning* est une méthode qui consiste à “éplucher” la forme jusqu'à obtenir un ensemble de points connectés d'un pixel de large qui conserve la topologie de la forme : le squelette. En d'autres termes, l'amincissement est une opération qui supprime itérativement les points simples (définition du point simple dans la sous-section 1.3.3) de la forme.

Il est nécessaire de noter que l'ordre de suppression des points est important quant au résultat obtenu. Effectivement, si l'on supprime les points simples dans l'ordre d'apparition du balayage de l'image (algorithme naïf 4), on obtient un squelette décentré comme présenté dans la figure 2.2.

Algorithme 4: Algorithme naïf d'amincissement

Données : l'image X

Résultat : le squelette Sk_X

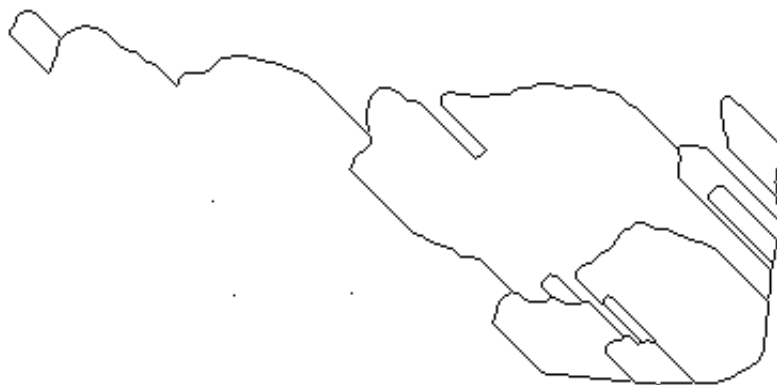
```

1 tant que  $fini = 0$  faire
2    $fini \leftarrow 1$ ;
3   pour chaque  $p$  dans l'ordre rasterscan faire
4     si  $X(p) = 1$  et simple( $p, X$ ) alors
5        $X(p) \leftarrow 0$ 
6        $fini \leftarrow 0$ 

```



(a) Forme binaire.



(b) Squelette de la forme binaire.

FIGURE 2.2 – Squelette calculé avec un algorithme d'amincissement naïf.

Il existe plusieurs méthodes de thinning, la plupart étant résumées dans [50].

Le principe de cette méthode consiste à examiner, dans un ordre prédéterminé, les points de contour de la forme afin de les supprimer. Ce processus peut s'effectuer dans un ordre lexicographique ou par suivi de contour [4, 53, 67].

Les points de contours sont ensuite marqués pour être supprimés après l'itération d'identification. Ces deux phases évitent de supprimer une branche complète du squelette dans une itération.

Pour définir sa simplicité, le "crossing number" d'Hilditch [45], Rutovitz [80] ou encore la "8-connexité" de Yokoi [98] sont généralement utilisés.

Les points de contour peuvent aussi être placés dans une queue [96] puis supprimés alternativement selon leur simplicité. L'algorithme continue cette phase jusqu'à ce qu'aucun point dans la queue ne soit simple.

L'avantage de la méthode d'amincissement ou de thinning, est qu'elle garantit l'homotopie du squelette grâce à la suppression des points simples. Cette méthode permet également d'obtenir un squelette fin puisque c'est une méthode qui supprime itérativement les points jusqu'à obtenir un ensemble de points non simples. Cependant cette méthode ne permet pas de reconstruire la forme initiale grâce au squelette. De plus, le squelette obtenu peut ne pas être bien centré dans la forme. C'est le cas pour l'amincissement naïf mais aussi pour les autres méthodes lorsque la forme à analyser a une épaisseur paire.

2.4 Squelettisation basée sur les distances

La méthode basée sur une carte de distance consiste à identifier des points clés sur la carte de distance et de les reconnecter pour obtenir le squelette.

La première étape est de calculer une carte de distance DT_X .

Dans ses travaux, Borgefors [24] considère un plan hexagonal pour calculer une distance en nid d'abeille.

Dans son cas, Arcelli [5] privilégie la distance d_4 .

Dans leur travaux, Thiel [92] et Attali [13] calculent des cartes de distances d_4 , d_8 , $d_{3,4}$ et $d_{5,7,11}$. Ils procèdent à un pré-traitement sur l'image pour supprimer les éventuelles irrégularités du bord de l'objet.

Ils calculent la ligne médiane de l'objet. Cette ligne est composée de points de l'axe médian et de points selles. Les points selles forment une crête reliant deux parties de la carte de distance où la valeur DT_X est plus grande [92]. Certains de ces points sont des points de l'axe médian mais les extrémités des crêtes ne sont pas forcément des points de l'axe médian.

Ces points selles, à identifier, remplissent au moins l'une des deux conditions suivantes :

- il y a 2 composantes 4-connectées de points ayant une valeur de distance inférieure à celle de p dans le voisinage \mathcal{N}_2 du point,
- il y a 2 composantes 8-connectées de points ayant une valeur de distance supérieure à celle de p dans le voisinage \mathcal{N}_2 du point.

La ligne médiane identifiée peut être déconnectée. Pour la reconnecter, il suffit d'identifier des points dits “*induits*” [13]. Ces points sont calculés suivant le gradient maximal de la carte de distance. L'identification de ces points est une étape délicate car des trous peuvent apparaître. Il est alors difficile de garantir l'homotopie de la forme.

Après identification de ces points, une phase de remplissage des éventuels trous est nécessaire pour préserver la topologie de la forme. Mais il se peut que le “squelette” obtenu ne soit pas fin. C'est pour cette raison qu'une phase de réduction de la ligne médiane est nécessaire. Dans [81], une phase consiste à marquer sur la carte de distance les points formant les trous et la phase suivante consiste à démarquer les points du squelette de telle sorte qu'il soit fin (le squelette résultant est l'ensemble des points de DT_X encore marqués).

Des éventuelles barbules peuvent ensuite être supprimées, cette phase se nomme l'ébarbulation ou l'élagage. Les barbules sont des morceaux du squelette qui apparaissent en présence d'irrégularités sur le contour de la forme.



FIGURE 2.3 – Exemple de forme (noir) présentant une irrégularité sur le bord, son squelette (blanc) présente alors une barbule.

Un critère d'élagage peut être défini pour supprimer ces barbules [12, 94]. Soit p un point situé à l'extrémité de la barbule et q un autre point situé en amont de p ($DT_X(q) > DT_X(p)$). Notons P et Q l'ensemble des points appartenant aux disques de centres p et q . Si le disque Q recouvre largement le disque P , alors la suppression du point p engendre peu de perte d'information sur la reconstruction du squelette. Le critère d'élagage permet de définir à quel degré ce disque P est recouvert par le disque Q : $DT_X(p) - DT_X(q) + d(p, q) < \theta$ où θ est le seuil défini par l'utilisateur.

Ce critère d'élagage est applicable aux différentes distances pondérées, mais au maximum 3 pondérations. Soit une distance $d_{a,b,c}$ le critère se définit de la façon suivante :

$$DT_X(p) - DT_X(q) + aH + bD + \sigma \min(H, D) < \theta \quad (2.1)$$

Avec H et D le nombre de déplacements horizontaux et diagonaux respectivement. Le seuil souvent utilisé est $\theta = 2a$. Enfin σ vaut 0 si $c = \infty$ comme pour d_4 , d_8 ou encore $d_{3,4}$, sinon $\sigma = c - b - a$.

Dans son travail de thèse, Gong [40] calcule une carte de distance octogonale (d_{oct}), détecte des points multiples et construit un chemin de connexion entre ces points. Comme le squelette obtenu n'est pas fin, il procède à une dernière phase d'amincissement. Pour remédier à ce problème de squelette non fin, il a également développé son algorithme sur une maille dérivée (sous section 5.4.1).

Le calcul de la carte de distance nécessite plusieurs balayages de l'image ainsi que la recherche de points de l'axe médian, points selles et leur reconnexion. Dans son travail de thèse [57], Nicolas Normand a proposé un algorithme de squelettisation basé sur une carte de distance en un balayage de l'image. Sa première étape a été de calculer une carte de distance en un balayage de l'image (voir la sous section 1.4.3). Dans une seconde partie, il a extrait l'axe médian et les points

selles. Enfin il a rétabli les chemins de connexion entre ces points en calculant les gradients de l'image.

L'avantage de cette méthode réside dans le fait que les points de l'axe médian sont préservés dans le squelette. Cela implique que le squelette est complètement centré dans la forme puisque les points de l'axe médian sont les centres des boules maximales incluses dans la forme. Cela veut aussi dire que ce type de squelette est reconstituable puisque l'information de distance a été enregistrée. Cependant ces squelettes ne sont pas, dans tous les cas, fins.

Lorsque la forme à analyser a une épaisseur paire, les points de l'axe médian forment un ensemble non fin. Un post-traitement amincissant le squelette est possible pour obtenir un squelette fin. Mais ce post-traitement nuira à la reconstruction de la forme. De plus, lors de la reconnexion des points de l'axe médian, cette méthode peut engendrer la formation locale de trous. La topologie de la forme n'est donc plus préservée. Il existe des post-traitements sur le squelette pour détecter ces faux trous et les remplir.

2.5 Squelettisation par approche hybride

Il existe des méthodes qui utilisent les principes des deux familles précédemment décrites (en section 2.3 et 2.4). L'une d'entre elles est un amincissement contrôlé par la distance, Distance Ordered Homotopic Thinning (DOHT) présentée par Pudney [71] et utilisée par Fouard en imagerie médicale [38].

Il présente deux méthodes d'amincissement contrôlé par la distance sur des images en $3D$. Les deux méthodes utilisent une règle différente de préservation des points :

- La première règle préserve l'extrémité de l'axe médian ou les contours des surfaces médianes,
- La seconde préserve le centre des boules maximales identifiées par la distance de chanfrein.

Dans la première boucle, il place les points de contour dans la queue. La figure 2.4 illustre un exemple où à la fin de la première boucle de l'algorithme, les points de contours ont été identifiés puis placés dans la queue. Ces points sont grisés sur la figure et les chiffres indiquent l'ordre de passage sur les pixels.

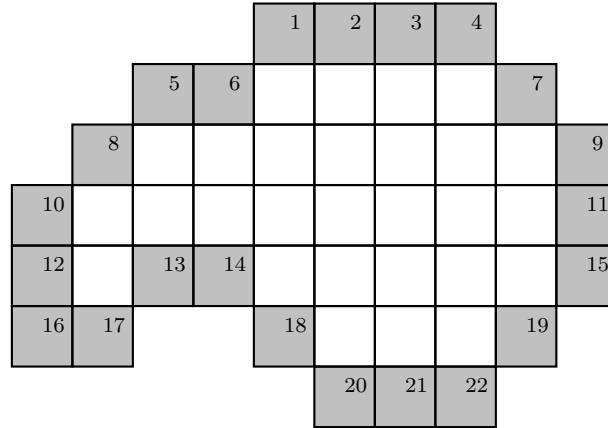


FIGURE 2.4 – Première étape de l'algorithme de Pudney.

Un label lui permet de savoir si le point est en attente. La boucle principale teste la suppression des points dans l'ordre de la queue. La figure 2.5 montre l'état des pixels après la suppression du premier point simple placé dans la queue.

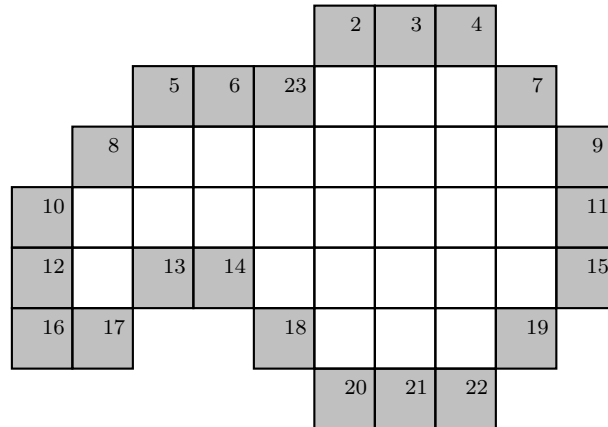


FIGURE 2.5 – Seconde étape de l'algorithme de Pudney.

Les points supprimables deviennent des points du fond et ses voisins ($\in X$) deviennent ainsi des points de contour placés à la fin de la queue. Si un point n'est pas supprimable, il est labelisé en tant que tel. L'algorithme se termine quand la queue est vide. C'est à dire lorsqu'il n'y a plus de point supprimable.

Cette méthode de squelettisation 3D peut être applicable au volume d'images sur lequel nous allons travailler par la suite.

Mais nous nous intéressons aux méthodes 2D séquentielles comme l'ont fait Svensson, Borgefors et Nyström [90].

Leur travail consiste à amincir la carte de distance d'un objet, couche par couche sans supprimer les points d'ancrage.

La première étape consiste à calculer une carte de distance. Leur algorithme fonctionne avec plusieurs types de distances, les résultats présentés sont calculés à partir de d_4 , d_8 , $d_{3,4}$, $d_{5,7,11}$ et d_e .

La seconde étape consiste à identifier les points de l'axe médian, nommé CMD pour "center of the maximal disks". Elles calculent ensuite un ensemble de CMD réduit : RCMDs (voir la section 1.6). L'ensemble des RCMDs sera considéré comme des points d'ancrage au squelette. L'utilisation de ce type de point d'ancrage permet au squelette d'être bien centré dans la forme et d'autre part de permettre une reconstruction parfaite de la forme.

Leur algorithme prévoit 2 balayages de l'image pour chaque valeur de distance. Pendant le premier balayage elles marquent les pixels p dont les propriétés sont les suivantes :

- $DT_X(p)$ correspond à la valeur de distance inspectée,
- p a un voisin de type \mathcal{N}_1 dans le fond,
- p est simple,
- p n'appartient pas au RCMDs.

Le second balayage supprime itérativement les points toujours simples. Pour calculer la simplicité d'un point elles utilisent le "crossing number" d'Hilditch [45].

Le squelette obtenu grâce à leur algorithme n'est pas fin puisque le RCMD n'est pas fin lui même et qu'il est conservé dans le squelette. C'est pourquoi elles pratiquent un post traitement qui consiste à amincir le squelette. Mais à cause de cette dernière phase le squelette n'est plus entièrement réversible. Il faut cependant souligner l'intérêt du calcul des RCMDs dans la reconstruction de la forme. Dans leur travaux elles comparent leurs reconstructions de forme à partir des squelettes avec comme point d'ancrage les CMDs et RCMDs. On constate alors qu'elles obtiennent une bien meilleure reconstruction lorsqu'elles utilisent les RCMDs comme point d'ancrage. Cela s'explique par la façon dont elles procèdent pour éliminer

des CMDs pour obtenir les RCMDs. En effet lorsqu'elles retirent un CMD, elles s'assurent préalablement que le point n'est pas pertinent dans la reconstruction de la forme ; alors que dans le cas où elles considèrent tous les CMDs comme point d'ancrage, le post-traitement qui consiste à amincir le squelette peut supprimer des points clés (nécessaires à la reconstruction).

2.6 Conclusion

Dans ce chapitre est présenté un puissant descripteur de forme : la squelettisation.

Elle permet d'extraire un sous ensemble de la forme avec certaines propriétés :

- restructibilité,
- l'homotopie,
- la finesse,
- la localisation : centré dans la forme.

Cependant ces propriétés ne peuvent pas toutes être obtenues.

Il existe plusieurs méthodes de squelettisation. Dans ce travail sont présentés trois types de méthodes 2D séquentielles :

- l'amincissement,
- la squelettisation basée sur les cartes de distances,
- la squelettisation par approche hybride.

Ces différentes méthodes permettent d'obtenir un squelette privilégiant certaines propriétés. Le tableau 2.1 résume les avantages et les inconvénients des méthodes de squelettisation présentées précédemment.

TABLE 2.1 – Récapitulatif des avantages et inconvénients des méthodes de squelettisation présentées.

Méthode	Homotope	Reconstructible	Centré	Fin
Thinning	+	–	–	+
Distances	–	+	+	–
Hybride	+	+	+	–

Il est nécessaire de mettre un ordre de priorité sur les propriétés du squelette que l'on souhaite conserver car cet ordre oriente le choix de la méthode à appliquer.

Conclusion

Dans cette partie, les notions de base de ce travail ont été introduites.

Dans un premier temps nous avons décrit l'espace discret. La géométrie discrète possède des propriétés qui permettent de décrire la relation entre les différents points de l'espace discret. Nous avons, par exemple, donné la définition de connexité d'un point par rapport à un autre. Plus globalement, nous nous sommes intéressés à la topologie d'une forme binaire. Cette forme étant composée de pixels noirs nous pouvons, par le biais d'algorithmes, décrire cet ensemble de points. Ainsi le nombre de composantes connexes nous indique le nombre de formes (ensemble de points qui ne touchent pas un autre ensemble de points) dans l'image.

Nous avons ensuite défini la notion de distance discrète. La mesure de distance en géométrie continue se réalise grâce à la distance euclidienne. Dans cette partie, nous avons exposé qu'il existe plusieurs distances discrètes dans le domaine discret. Dans ce travail nous utiliserons les distances d_4 se référant au voisinage $\mathcal{N}1$, d_8 se référant au voisinage $\mathcal{N}2$ et la distance à séquence de voisinage d_{ns} . Cette dernière utilise une séquence des deux voisinages $\mathcal{N}1$ et $\mathcal{N}2$. À noter que lorsque la séquence de voisinage alterne simultanément $\mathcal{N}1$ et $\mathcal{N}2$, on l'appelle la distance octogonale d_{oct} .

Cette définition de distance permet de calculer la carte de distance d'une image. La carte de distance DT_X attribue à chaque pixel sa valeur de distance au fond de la forme.

La notion d'axe médian MA_X a ensuite été introduite. C'est un outil de compression d'image performant car il est constitué de tous les centres des boules maximales complètement incluses dans la forme. Mais l'axe médian ne décrit pas la forme. Pour cela nous avons présenté, dans la deuxième section de cette partie, le squelette et des algorithmes de squelettisation permettant de le calculer.

Le squelette est une représentation efficace de la forme qui représente la ligne fine (au mieux 1 pixel de large) centrée dans la forme. Cette représentation est un excellent descripteur de forme puisque le squelette conserve la topologie de l'objet. Les méthodes présentées se situent dans le domaine 2D et sont des méthodes

séquentielles. On compte trois types de méthodes :

- L'amincissement,
- Méthode basée sur les cartes de distance,
- Méthode hybride.

Ces trois méthodes privilégient différentes propriétés du squelette. Effectivement, toutes les propriétés du squelette (homotopie, finesse, restructibilité, centré dans la forme) ne peuvent pas être réunies dans un même squelette.

Alors que l'amincissement épluche la forme jusqu'à obtenir le squelette, la méthode basée sur les cartes de distance identifie des points spéciaux (points de l'axe médian, points selle) et procède à leur reconnexion. Comme son nom l'indique, la méthode hybride est un mélange des deux premières méthodes : c'est un amincissement contrôlé par la distance.

Ces traitements (calcul de la carte de distance ou encore la squelettisation) nécessitent plusieurs balayages de l'image pour obtenir une information centrée dans la forme.

Dans la suite de ce travail nous allons mettre en œuvre des algorithmes de calcul (DT_X , MA_X , squelettisation) requérant un balayage unique de l'image. Ces algorithmes serviront à extraire des informations sur la microarchitecture osseuse.

Deuxième partie

La squelettisation en un balayage

Introduction

La reconnaissance de forme, en analyse d'image, désigne la reconnaissance des éléments présents dans une image. On reconnaît des éléments par leur forme, leur taille, leur couleur, leur texture, leur structure.... C'est par cette dernière caractéristique que nous allons aborder l'analyse d'image.

Le but ce travail sera d'analyser l'architecture de l'os grâce à des images tomographiques de la mâchoire. L'étude de cette architecture va nous amener à extraire des caractéristiques quantitatives suffisantes pour caractériser l'architecture osseuse. La squelettisation est une méthode d'analyse d'image qui permet de mettre en avant les caractéristiques structurales de l'objet.

La figure II.1 montre les différents traitements opérés sur l'image. La première phase consiste à acquérir l'image avec un micro scanner puis la seconde étape consiste à extraire la région d'intérêt et à binariser l'image. Ces deux étapes seront expliquées dans la partie III. La troisième phase, celle que nous décrirons dans cette partie, consiste à extraire le squelette de la forme, ici l'os. Les quatrième et cinquième étapes consistent à extraire les caractéristiques du squelette et de les analyser. Ces deux dernières phases seront expliquées dans la partie III.

FIGURE II.1 – Schéma des traitements mis en œuvre.

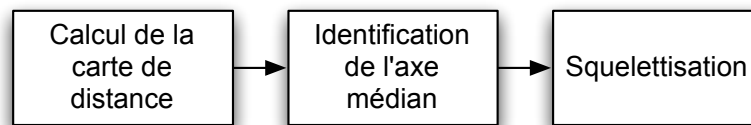


Dans cette partie est décrite une nouvelle méthode de squelettisation contrôlée par la distance. La distance est une mesure quantitative importante pour mesurer la qualité de l'os. Cet algorithme de squelettisation fonctionne en un balayage unique de l'image. Ce qui n'est pas le cas pour les méthodes séquentielles existantes de squelettisation qui nécessitent plusieurs balayages de l'image pour calculer la

carte de distance et pour opérer la fonction simplicité sur chaque pixel. Prenons l'exemple d'un algorithme d'amincissement. Le nombre de balayages de l'image dépend directement de la largeur de la forme à squelettiser, soit le rayon maximal d'un disque complètement inclus dans la forme. Pour un algorithme basé sur la carte de distance, deux balayages sont nécessaires pour calculer la carte de distance puis plusieurs balayages (en fonction de la forme) pour squelettiser.

La figure II.2 montre les différentes étapes du processus de squelettisation.

FIGURE II.2 – Schéma du processus de squelettisation.



Cette partie se compose naturellement de trois chapitres.

Le premier chapitre décrira le calcul d'une carte de distance (définition 3.3) lisible en un balayage, on l'appelle carte de distance "asymétrique". Dans son travail de thèse, Nicolas Normand [57] calcule également une carte de distance asymétrique à séquence de voisinages, pas nécessairement périodique. Elle est calculée avec des tables de correspondances pré-calculées.

Dans ce travail, le calcul de la carte de distance asymétrique se produit à la volée avec l'aide des séquences de Beatty.

Nous montrons qu'une carte de distance asymétrique (calculée avec des disques décentrés) est équivalente à la carte de distance symétrique au sens où l'information qu'elle porte est identique, seule change sa localisation. Nous donnons également un algorithme pour recentrer cette carte, en place et en temps constant par pixel. Bien que la carte de distance ne soit pas explicitement recentrée pour la squelettisation, l'algorithme de recentrage sert à ordonnancer la suppression de pixels simples.

Le second chapitre expliquera le processus d'identification des points de l'axe médian à partir de la carte de distance asymétrique. Nous montrons qu'il est équivalent à l'axe médian calculé sur une carte de distance centrée.

Enfin, le calcul du squelette sera présenté dans le troisième chapitre. Des travaux précédents [57] ont montré des méthodes de squelettisation basées sur une carte de distance asymétrique. Comme toutes les méthodes de squelettisation basées sur les cartes de distance, elle rompt l'homotopie lors de l'extraction des points spéciaux (selles et axe médian) puis la rétablit lors de la reconnection de ces points selon le gradient de l'image.

Dans ce travail, nous proposons un algorithme de squelettisation de type hybride. La méthode de squelettisation hybride garantit l'homotopie du squelette, préserve l'information de distance, permet la reconstruction du squelette et le squelette produit est centré dans la forme. L'algorithme de squelettisation en un balayage présenté dans ce travail calcule séquentiellement un squelette avec une empreinte mémoire minimale.

Dans chacune de ces parties seront présentés indépendamment les résultats correspondant à la carte de distance, l'axe médian puis le squelette. Ces résultats sont produits à partir d'une base d'images composée de 57 images binaires de tailles et de formes variables. L'ordinateur sur lequel les algorithmes ont été testés est muni d'un processeur 2.4 Ghz Intel Core 2 Duo.

Calcul de la carte de distance translatée

3.1 Introduction

La carte de distance, calculée à l'aide de l'équation 3.1, attribue à chaque pixel sa distance depuis le pixel du fond le plus proche.

$$DT_X(p) = \min\{d(q, p) : q \in \mathbb{Z}^2 \setminus X\}; \quad (3.1)$$

Comme la distance est symétrique, cela revient également à dire que la carte de distance attribue à chaque pixel la distance du fond à ce pixel. La figure 3.1 rappelle un exemple de carte de distance.

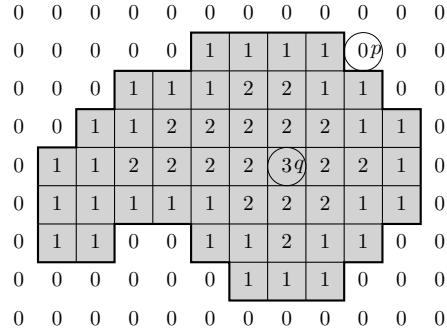
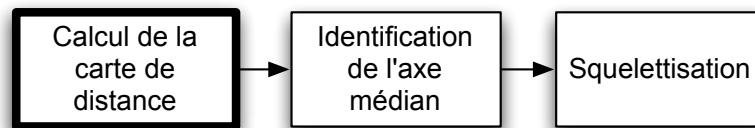


FIGURE 3.1 – Exemple de carte de distance d_8 .

Sur cette figure, la distance séparant le point p du point q est la même que la distance séparant le point q du point p car toute distance est toujours symétrique. Cette notion est importante pour la suite de ce chapitre car nous allons calculer

une carte de distance asymétrique.

FIGURE 3.2 – Schéma du processus de squelettisation.



Comme le montre le schéma de la figure 3.2, la première étape de notre méthode est d’obtenir une carte de distance qui nous permettra de contrôler l’amincissement. Dans ce chapitre nous nous intéresserons donc à définir et calculer une carte de “distance” lisible et calculable en un balayage de l’image. Les pixels seront amincis dans l’ordre de leur valeur dans la carte de distance DT_X . Mais au lieu de calculer DT_X explicitement, on en calcule une version décentrée notée DT'_X , à la volée. DT'_X correspond à une distance “asymétrique” notée d' . Elle nous permettra par la suite de calculer l’axe médian puis le squelette.

Une première section 3.2 définit les distances à poids variables. Dans la deuxième section (3.3) nous allons calculer DT'_X en un seul balayage de l’image. Pour cela nous nous appuierons sur l’étude de la propagation de l’information dans l’image décrite dans le chapitre 1.7. Pour calculer cette carte de “distance” en un seul balayage de l’image, nous allons translater les centres des disques.

La troisième section (3.4) met en évidence l’équivalence entre DT'_X et DT_X quelle que soit la façon de calculer cette dernière. Nous montrerons également comment, à partir de DT'_X , on peut retrouver DT_X et nous discuterons de cette procédure en termes d’efficacité algorithmique.

Les résultats obtenus grâce à cet algorithme seront présentés dans une quatrième section (3.5).

3.2 Distances à poids variables

Dans ce travail nous considérons une distance basée sur les chemins, la fonction qui associe à chaque couple de points (p,q) , le coût minimal c d’un chemin allant

de p à q .

$$c_{\vec{v}}(r) = \begin{cases} r + 1 & \text{si } \vec{v} \in \mathcal{N}_{B(r)} \\ r + \infty & \text{sinon} \end{cases} \quad (3.2)$$

3.3 Calcul de DT'_X

La carte de distance attribuée à chaque pixel, la distance de ce pixel depuis le pixel du fond le plus proche (définition de la transformation en distance 3.3). C'est-à-dire que pour chaque pixel il existe un chemin qui relie ce point au fond dont la distance est représentée sur ce pixel (exemple figure 3.1).

Dans ce travail nous mettons en œuvre des calculs de distances asymétriques correspondant aux distances d_4 , d_8 et d_{NS} (définies précédemment dans la section 1.4). Afin de généraliser notre méthode, nous présenterons le calcul d'une carte de distance avec une distance à séquence de voisinages. Les distances d_4 et d_8 sont respectivement des séquences de $B = \{1, 1, 1, 1, \dots\} = (\bar{1})$ et $B = \{2, 2, 2, 2, \dots\} = (\bar{2})$.

3.3.1 Translation des centres des disques

Dans le but de calculer la carte de distance avec un seul balayage de l'image, nous appliquons un vecteur de translation \vec{t} au voisinage \mathcal{N} . Cette translation déplace le centre du disque au dernier point lu dans le voisinage. De ce fait, lors du balayage de l'image, le voisinage du point à calculer aura déjà été visité ce qui implique que l'on peut attribuer la valeur de distance au point à calculer.

D'après les travaux de Verwer [95], il existe deux modes de calcul d'une carte de distance. Lorsque l'on se place sur le point courant p_0 et que l'on calcule sa distance depuis le fond en examinant son voisinage précédemment calculé, on utilise le "*read formalism*". Dans le cas du "*write formalism*", lors du calcul du point courant p_0 on propage l'information de distance à ses voisins postérieurs.

Deux représentations de la translation de voisinage sont possibles en accord avec

le formalisme de Verwer.

Dans le premier cas, on propage l'information de distance sur l'ensemble du voisinage à partir d'un point sans revenir en arrière (*"write formalism"*). Cependant on ne calcule pas la valeur de distance du point courant, on la réajuste. La figure 3.3 montre les vecteurs de déplacements de l'information.

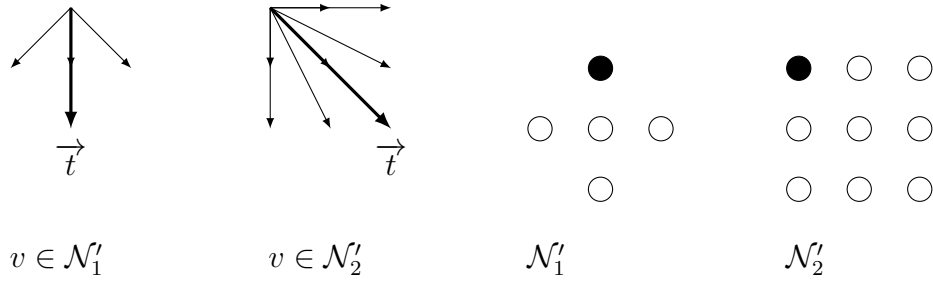


FIGURE 3.3 – Propagation de l'information dans \mathcal{N}'_1 et \mathcal{N}'_2 .

Cette représentation met en évidence la propagation des chemins.

Dans le second cas, on se sert du voisinage du point courant déjà calculé pour connaître la valeur de distance de ce point (*"read formalism"*). La figure 3.4 montre les vecteurs de déplacements de l'information.

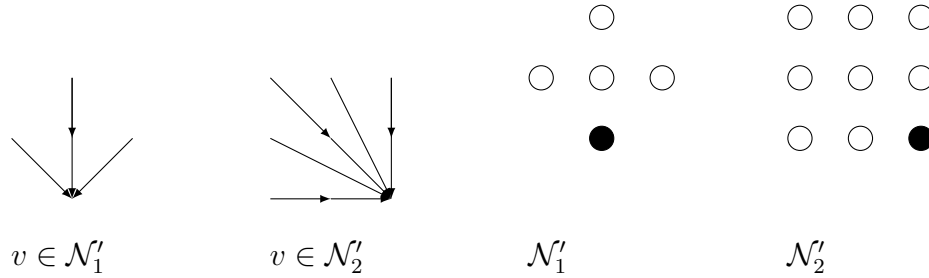


FIGURE 3.4 – Propagation de l'information dans les voisinages symétriques \mathcal{N}'_1 et \mathcal{N}'_2 .

Cette représentation permet de voir que lorsqu'on calcule un point qui est le centre translaté du disque, son voisinage a déjà été visité.

La figure 3.5 montre les différents vecteurs de translation des voisinages \mathcal{N}_1 et \mathcal{N}_2 . Elle montre également tous les vecteurs possibles $v \in \mathcal{N}'_1 \cap \mathcal{N}'_2$ (g) mais aussi

les vecteurs utilisés dans toute la séquence $B(r) \ v \in \mathcal{N}'_1 \cup \mathcal{N}'_2$ (h). C'est-à-dire quel que soit le type de voisinage utilisé.

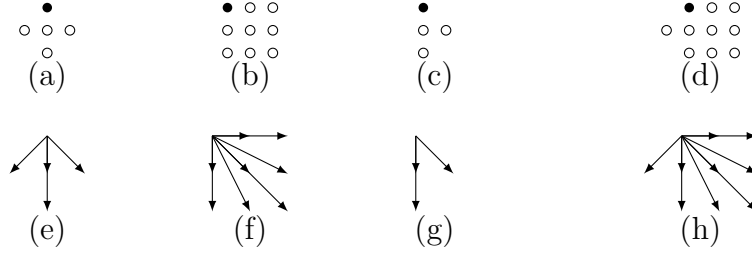


FIGURE 3.5 – Voisinages utilisés pour calculer DT'_X et leur vecteurs de translation associés.

3.3.2 La séquence de voisinage

La carte de distance attribue à chaque point p sa distance depuis le pixel du fond le plus proche q :

$$DT_X(p) = \min\{d(q, p) : q \in \mathbb{Z}^2 \setminus X\}; \quad (3.3)$$

Exprimée d'une autre façon, la carte de distance attribue à chaque point p la valeur du rayon r du disque maximal de centre p :

$$DT_X(p) = \max \{r : \check{D}_{<}(p, r) \subset X\} \quad (3.4)$$

Rappelons la définition d'un disque (équation 1.10) :

$$D_{\leq}(p, r) = \{q : d(p, q) \leq r\} \quad (3.5)$$

Nous appliquons à présent un vecteur de translation $\vec{t}(r)$ aux disques.

$$\vec{t}(r) = \mathbf{1}_B(r)\vec{t}_1 + \mathbf{2}_B(r)\vec{t}_2; = (\mathbf{1}_B(r), r) \quad (3.6)$$

Avec \vec{t}_1 et \vec{t}_2 les vecteurs de translation appliqués respectivement dans \mathcal{N}_1 et \mathcal{N}_2 .

La définition du disque translaté est noté D' :

$$D'(p, r) = D(p + \vec{t}(r), r) \quad (3.7)$$

Depuis les équations 3.4 et 3.7, la distance asymétrique peut s'écrire :

$$DT'_X(p) = \max \{r : D(p + \vec{t}_r, r) \subset X\} \quad (3.8)$$

Soit \check{D} le symétrique du disque D :

$$\check{D}(p, r) = \{q : d(q, p) \leq r\} \quad (3.9)$$

D'après \check{D} , notons \check{D}' le symétrique du disque translaté :

$$\check{D}'(p, r) = \check{D}(p - \vec{t}(r), r) \quad (3.10)$$

La figure 3.6 présente les suites de disques, translatés avec $\vec{t}_1 = (0, 1)$ et $\vec{t}_2 = (1, 1)$, correspondants aux distances d_4 , d_8 et d_{oct} . La figure 3.7 présente les suites de disques symétriques correspondants aux distances d_4 , d_8 et d_{oct} .

Soit la séquence B , correspondant à une distance octogonale, suivante :

TABLE 3.1 – Suite de Beatty

r	1	2	3	4	5	6	7	8	9	10	11	12
$B(r)$	1	2	1	2	1	2	1	2	1	2	1	2
$\mathbf{1}_B(r)$	1	1	2	2	3	3	4	4	5	5	6	6
$\mathbf{2}_B(r)$	0	1	1	2	2	3	3	4	4	5	5	6
$\mathbf{1}_B^\dagger(r)$	0	2	4	6	8	10	12	14	16	18	20	22
$\mathbf{2}_B^\dagger(r)$	1	3	5	7	9	11	13	15	17	19	21	23

Rappelons (voir 1.4.3) que r correspond à la valeur du rayon du disque, $B(r)$ correspond au type de voisinage (\mathcal{N}_1 ou \mathcal{N}_2) utilisé au rayon r .

La valeur $\mathbf{1}_B(r)$ correspond au nombre de 1 apparus dans la séquence B au rayon r . Par exemple, $\mathbf{1}_B(8) = 4$ veut dire qu'il y a 4 voisinages de type \mathcal{N}_1 apparus dans le début de la séquence jusqu'à $B(8)$. De la même manière, $\mathbf{2}_B(r)$ correspond au nombre de 2 apparus dans la séquence B au rayon r .

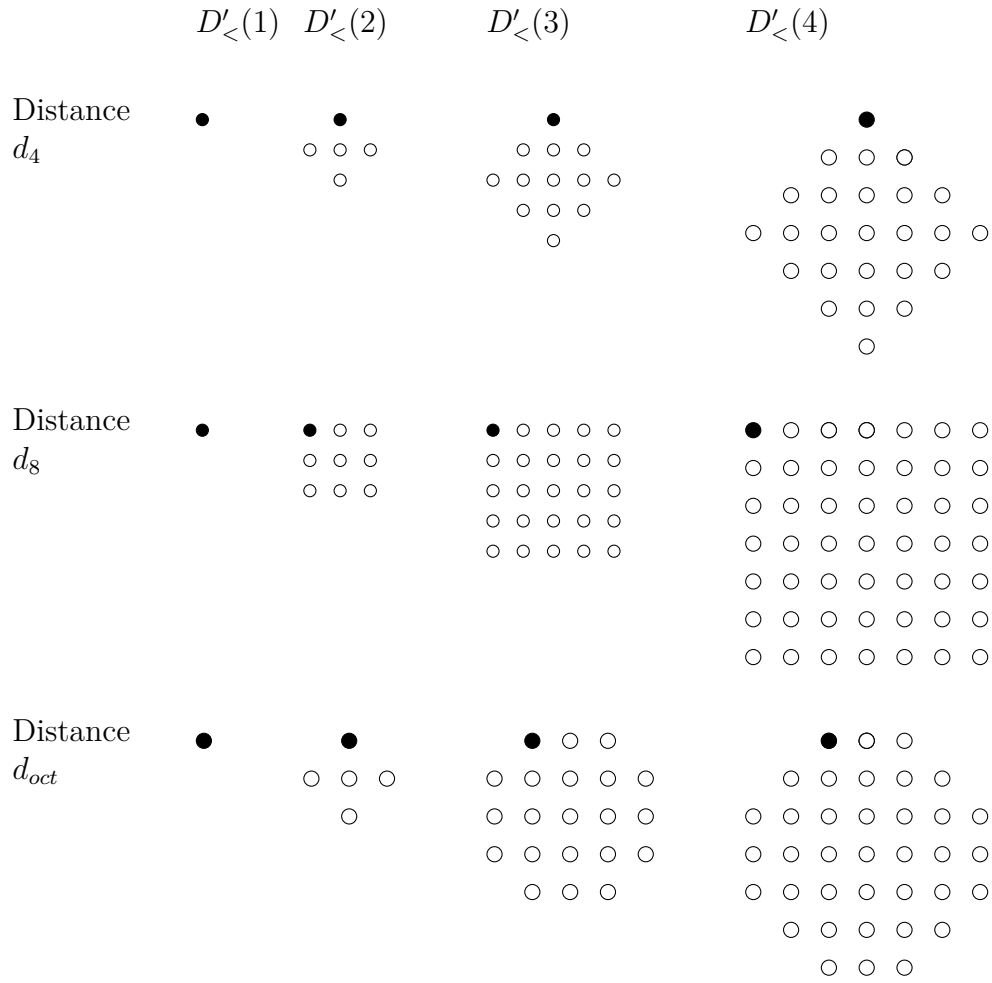


FIGURE 3.6 – Suites de disques \mathcal{N}_1 et \mathcal{N}_2 pour les rayons 1, 2, 3 et 4. Le point noir correspond au centre du disque.

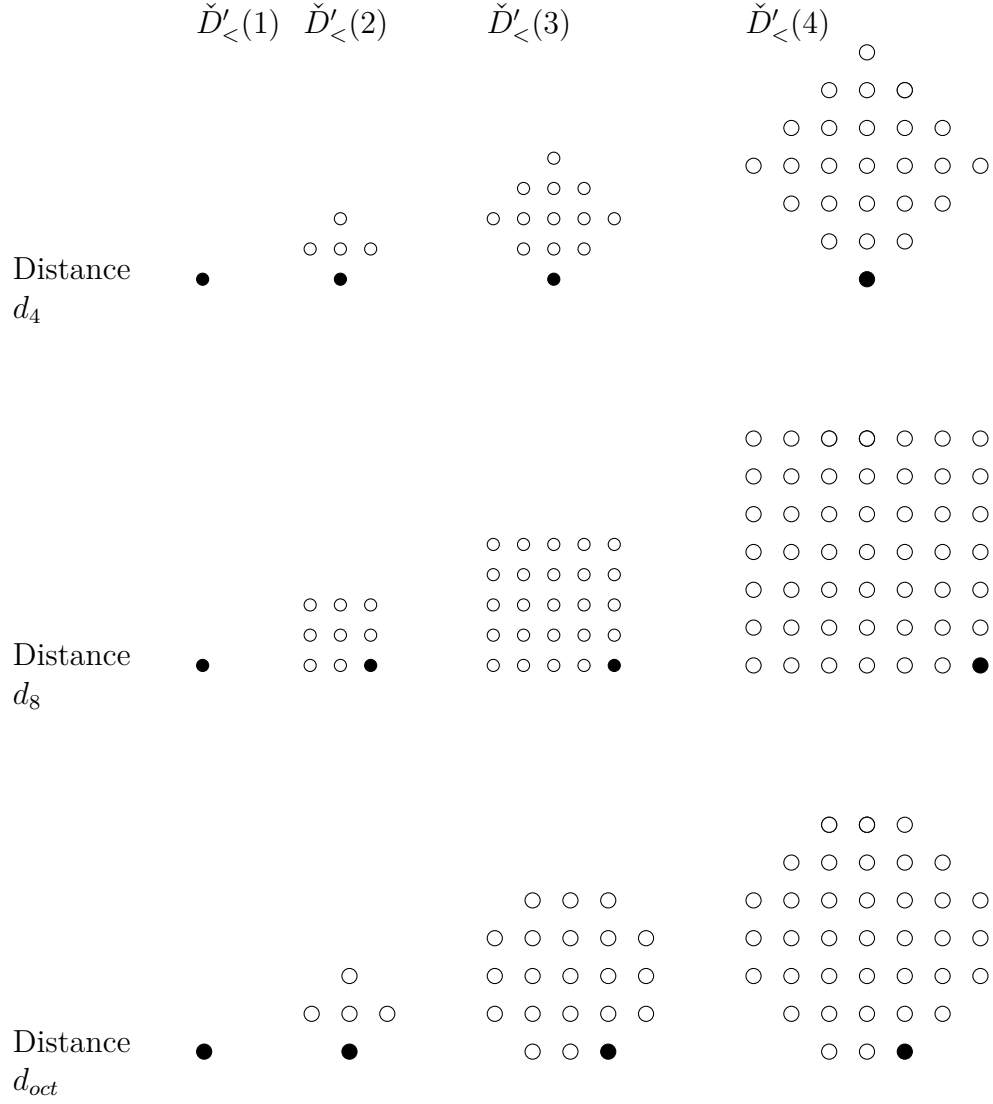


FIGURE 3.7 – Suites de disques symétriques \mathcal{N}_1 et \mathcal{N}_2 pour les rayons 1, 2, 3 et 4. Le point noir correspond au centre du disque.

La valeur $\mathbf{1}_B^\dagger(r)$ représente le nombre de valeurs dans $\mathbf{1}_B$ strictement inférieures au rayon r [49]. Par exemple, $\mathbf{1}_B^\dagger(7) = 12$ veut dire qu'il y a 12 valeurs strictement inférieures au rayon $r = 7$ dans la suite $\mathbf{1}_B$. De même $\mathbf{2}_B^\dagger(r)$ représente le nombre de valeurs dans $\mathbf{2}_B$ strictement inférieures au rayon r . On remarque qu'avec une séquence $B = \{1, 2\}$ les deux suites $\mathbf{1}_B^\dagger$ et $\mathbf{2}_B^\dagger$ sont complémentaires, ce qui n'est pas forcément vrai pour une autre séquence de voisinage.

D'après nos précédents travaux [60], on sait que $\mathbf{1}_B^\dagger(\mathbf{1}_B(r) + 1) + 1$ nous indique l'indice de la prochaine augmentation de la suite $\mathbf{1}_B(r)$ après r . Par exemple si l'on souhaite connaître, à la valeur $r = 5$, où se situe la prochaine augmentation de la suite $\mathbf{1}_B(r)$ on applique l'équation suivante :

$$\mathbf{1}_B^\dagger(\underbrace{\mathbf{1}_B(5)}_3 + 1) + 1 = \mathbf{1}_B^\dagger(\underbrace{4}_6) + 1 = 7 \quad (3.11)$$

Le résultat, 7, est l'indice de la suite $\mathbf{1}_B(r)$ dans laquelle il y a une augmentation. Car on passe de $\mathbf{1}_B(6) = 3$ à $\mathbf{1}_B(7) = 4$.

Les sommes cumulées des fonctions caractéristiques des vecteurs dans $\mathcal{N}'_1 \setminus \mathcal{N}'_2$, $\mathcal{N}'_2 \setminus \mathcal{N}'_1$ et $\mathcal{N}'_1 \cap \mathcal{N}'_2$ sont respectivement $\mathbf{1}_B$, $\mathbf{2}_B$ et la valeur constante 1 résultent des coûts de déplacement minimaux suivant :

$$\hat{C}_v(r) = \begin{cases} \hat{C}_v^1(r) = \mathbf{1}_B^\dagger(\mathbf{1}_B(r) + 1) + 1 & \text{si } v \in \mathcal{N}'_1 \text{ et } v \notin \mathcal{N}'_2 \\ \hat{C}_v^2(r) = \mathbf{2}_B^\dagger(\mathbf{2}_B(r) + 1) + 1 & \text{si } v \notin \mathcal{N}'_1 \text{ et } v \in \mathcal{N}'_2 \\ \hat{C}_v^{12}(r) = r + 1 & \text{si } v \in \mathcal{N}'_1 \text{ et } v \in \mathcal{N}'_2 \end{cases} \quad (3.12)$$

Les distances d_4 , d_8 et d_{oct} sont des distances de voisinage particulier. Pour la distance octogonale d_{oct} , où $B = (\overline{1, 2})$, les coûts de déplacement minimaux deviennent :

$$\hat{C}_v(r) = \begin{cases} \hat{C}_v^1(r) = 2 \left\lceil \frac{r+1}{2} \right\rceil + 1 & \text{si } v \in \mathcal{N}'_1 \text{ et } v \notin \mathcal{N}'_2 \\ \hat{C}_v^2(r) = 2(\mathbf{2}_B(r) + 1) = 2 \left\lceil \frac{r}{2} \right\rceil + 2 & \text{si } v \notin \mathcal{N}'_1 \text{ et } v \in \mathcal{N}'_2 \\ \hat{C}_v^{12}(r) = r + 1 & \text{si } v \in \mathcal{N}'_1 \text{ et } v \in \mathcal{N}'_2 \end{cases} \quad (3.13)$$

Concernant la distance d_4 , la séquence de voisinage associée est $B = (\bar{1})$ et les coûts de déplacement minimaux sont :

$$\hat{C}_v(r) = \begin{cases} \mathbf{1}_B^\dagger(\mathbf{1}_B(r) + 1) + 1 = \mathbf{1}_B^\dagger(r + 1) + 1 = r + 1 & \text{si } v \in \mathcal{N}'_1 \\ \infty & \text{si } v \in \mathcal{N}'_2 \end{cases} \quad (3.14)$$

De la même manière, la distance d_8 possède une séquence $B = (\bar{2})$ et les coûts de déplacement minimaux sont :

$$\hat{C}_v(r) = \begin{cases} \mathbf{2}_B^\dagger(\mathbf{2}_B(r) + 1) + 1 = \mathbf{2}_B^\dagger(r + 1) + 1 = r + 1 & \text{si } v \in \mathcal{N}'_2 \\ \infty & \text{si } v \in \mathcal{N}'_1 \end{cases} \quad (3.15)$$

3.3.3 Algorithme et exemple

L'algorithme 5 se déroule en un seul balayage de l'image. Pour tous les points de l'image dans un ordre "raster scan" (de gauche à droite et de bas en haut) si le point courant est dans le fond de l'image alors la valeur dans la carte de distance DT'_X vaut zéro. Sinon on calcule le coût minimal de déplacement pour tous les

vecteurs v .

Algorithme 5: Carte de distance asymétrique.

Données : X l'ensemble des points de la forme

Données : \mathcal{N} : le voisinage

Données : $\hat{C}_{\vec{v}}$: coût minimal absolu de déplacement

Résultat : DT'_X : carte de distance asymétrique de X

```

1 pour chaque  $p$  dans l'ordre raster scan faire
2   si  $p \notin X$  alors
3      $DT_X(p) \leftarrow 0$ 
4   sinon
5      $r \leftarrow \infty$ 
6     pour chaque  $\vec{v}$  dans  $\mathcal{N}$  faire
7        $r \leftarrow \min \{r; \hat{C}_{\vec{v}}(DT_X(p - \vec{v}))\}$ 
8      $DT_X(p) \leftarrow r$ 

```

Pour mieux comprendre ce qui se passe dans cette fonction, mettons-y un point d'arrêt. La figure 3.8 montre le résultat de la carte de distance au point d'arrêt. La majeure partie de la carte a déjà été calculée et nous nous arrêtons sur un point. En utilisant l'équation précédente et les vecteurs associés nous déterminons la valeur de distance au point non calculé.

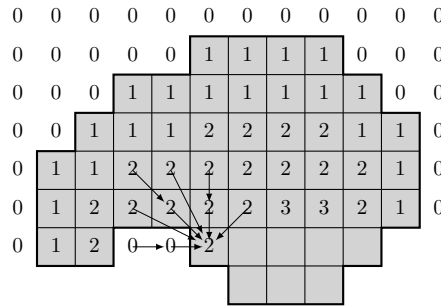


FIGURE 3.8 – Point d'arrêt dans le calcul de la carte de distance asymétrique d_{oct} .

Dans ce cas de figure nous avons des vecteurs dans \mathcal{N}'_1 , \mathcal{N}'_2 mais aussi dans les deux en même temps. Le coût de déplacement associé au vecteur dans \mathcal{N}'_1 avec

une valeur de distance de 2 dans DT'_X est de :

$$\mathbf{1}_B^\dagger(\mathbf{1}_B(2) + 1) + 1 = \mathbf{1}_B^\dagger(1 + 1) + 1 = 3$$

Le coût de déplacement associé aux vecteurs dans \mathcal{N}'_2 avec une valeur de distance de 2 dans DT'_X est de :

$$\mathbf{2}_B^\dagger(\mathbf{2}_B(2) + 1) + 1 = \mathbf{2}_B^\dagger(0 + 1) + 1 = 2$$

Puis avec une valeur de distance de 0 dans DT'_X est de :

$$\mathbf{2}_B^\dagger(\mathbf{2}_B(0) + 1) + 1 = \mathbf{2}_B^\dagger(0 + 1) + 1 = 2$$

Enfin, le coût de déplacement associé aux vecteurs appartenants aux deux voisinages est :

$$r + 1 = 3$$

Le coût de déplacement minimal vaut donc 2 comme l'indique l'exemple.

3.4 Équivalence et réversibilité de DT_X et DT'_X

Nous venons de calculer une carte de distance avec un balayage de l'image. Nous allons montrer que cette carte de “distance” asymétrique DT'_X est équivalente à une carte de distance centrée DT_X .

La figure 3.9 présente deux cartes de distance : DT_X étant calculée selon la littérature et DT'_X selon notre algorithme en un balayage. Cet exemple met en évidence que sur la carte de “distance” asymétrique DT'_X , le centre du disque de rayon 3 encerclé se trouve décentré par rapport à la carte de distance DT_X .

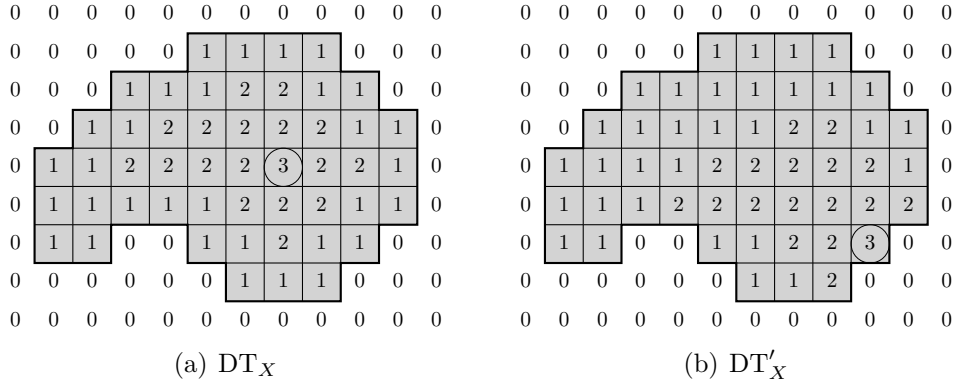


FIGURE 3.9 – Exemple de carte de distance d_8 DT_X et son homologue asymétrique DT'_X .

Cette section montre, dans un premier temps dans la sous-section 3.4.1, l'équivalence de ces deux cartes de distances. Puis dans la sous-section 3.4.2, sera présenté un algorithme qui permet de calculer DT_X à partir de DT'_X .

3.4.1 Équivalence entre DT_X et DT'_X

Soit DT'_X la “distance” tradatée de DT_X et $D'(p, r)$ le disque dont le centre a été tradaté de $\vec{t}_{(r)}$.

Si la valeur dans la carte de distance au point p est supérieure ou égale au rayon r du disque, cela veut dire qu'il existe une boule centrée sur le point p et de rayon r qui est incluse dans la forme.

$$DT_X(p) \geq r \Leftrightarrow D(p, r) \subset X \quad (3.16)$$

De même pour une valeur de la carte de “distance” tradatée au point p supérieure ou égale au rayon r , il existe un disque dont le centre tradaté est p avec un rayon de r qui est inclus dans la forme.

$$DT'_X(p) \geq r \Leftrightarrow \check{D}'(p, r) \subset X \quad (3.17)$$

Ce qui vient à dire que les disques $D'(p, r)$ et $D(p, r)$ sont homologues si on

admet que la valeur de distance au centre du disque tradatée est égal à la valeur de distance du centre en prenant en compte sa translation.

$$D'(p, r) = D(p + \overrightarrow{t(r)}, r) \quad (3.18)$$

et donc

$$DT'_X(p) \geq r \Leftrightarrow D_{<}(p + \overrightarrow{t(r)}, r) \subseteq X \Leftrightarrow DT_X(p + \overrightarrow{t(r)}) \geq r \quad (3.19)$$

On retrouve alors l'équivalence entre la carte de distance et la carte de distance tradatée.

$$DT_X(p) = r \Leftrightarrow \begin{cases} DT'_X(p + \overrightarrow{t(r)}) \geq r \\ DT'_X(p + \overrightarrow{t(r+1)}) \leq r \end{cases} \quad (3.20)$$

3.4.2 Réversibilité : Calcul de DT_X à partir de DT'_X

Nous venons de voir que DT'_X était équivalent à DT_X . Nous pouvons donc calculer DT_X à partir de DT'_X . Pour cela il suffit de connaître le vecteur de translation $\overrightarrow{t(r)}$ et la valeur du rayon du disque r .

La figure 3.10 illustre un exemple de carte de distance tradatée. Dans 3.10(a) le vecteur représenté est utilisé lors de la création de la carte de distance tradatée. Dans 3.10(b) on utilise ce même vecteur afin de retrouver la valeur dans la carte de distance.

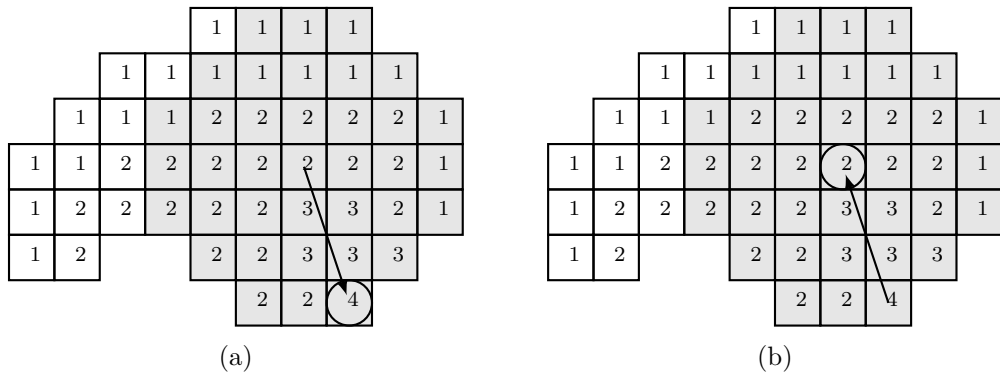


FIGURE 3.10 – Exemple de vecteur de translation pour une distance d_{oct} .

Lorsque l'on souhaite détranslater la carte de distance asymétrique DT'_X , il faut nécessairement connaître en chaque point le vecteur de translation qui a été utilisé lors du calcul de la carte. En d'autres termes, il faut connaître le type de voisinage $\mathcal{N}1$ ou $\mathcal{N}2$ qui a servi pour calculer le point.

Algorithme 6: Calcul de DT_X à partir de DT'_X

Données : Carte de distance translatée DT'_X

Résultat : Carte de distance DT_X

```

1  $r_0 \leftarrow DT'_X(p')$ ;
2  $r_1 \leftarrow \max\{1, DT'_X(p' + \vec{t}_1)\}$ ;
3  $r_2 \leftarrow \max\{1, DT'_X(p' + \vec{t}_2)\}$ ;
4 pour  $r \in [r_1, r_0]$  faire
5   | si  $b(r)=1$  alors
6   |   |  $DT_X(p' - \vec{t}(r)) \leftarrow r$ ;
7   | fin
8 fin
9 pour  $r \in [r_2, r_0]$  faire
10  | si  $b(r)=2$  alors
11  |   |  $DT_X(p' - \vec{t}(r)) \leftarrow r$ ;
12  | fin
13 fin
```

La ligne 6 de l'algorithme représente le fait que le disque $D_<$ de centre $p' - \vec{t}(r)$ et de rayon r est un disque contenu dans la forme X :

$$D(p' - \vec{t}(r), r) \in X \quad (3.21)$$

3.5 Résultats

La figure 3.11 montre trois images binaires utilisées pour calculer les cartes de distances classiques (représentées dans les figures 3.12, 3.14 et 3.16) et translatées (représentées dans les figures 3.13, 3.15 et 3.17) suivant les distances d_4 , d_8 et d_{oct} .

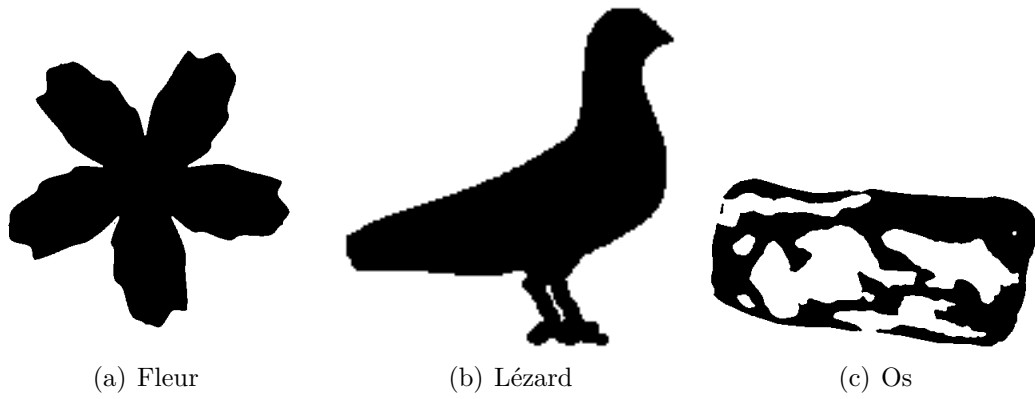


FIGURE 3.11 – Images originales binaires.

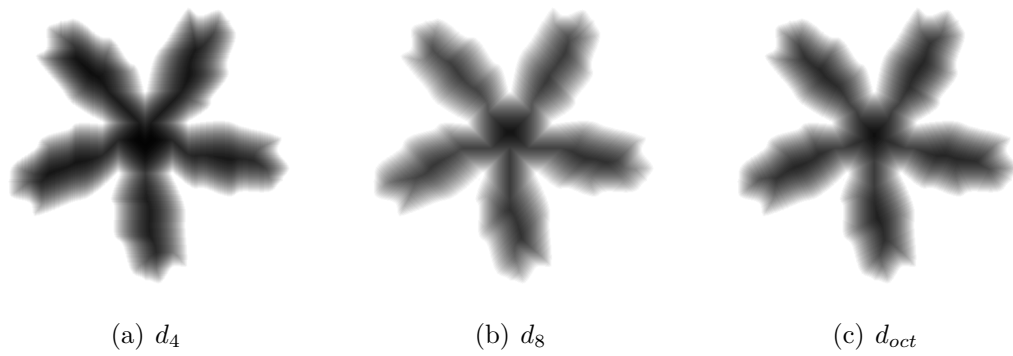


FIGURE 3.12 – Cartes de distance de la fleur.

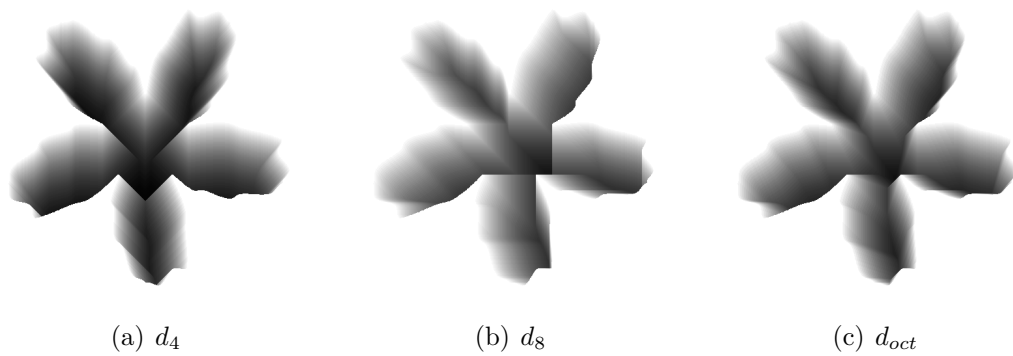


FIGURE 3.13 – Cartes de distance translatées de la fleur.

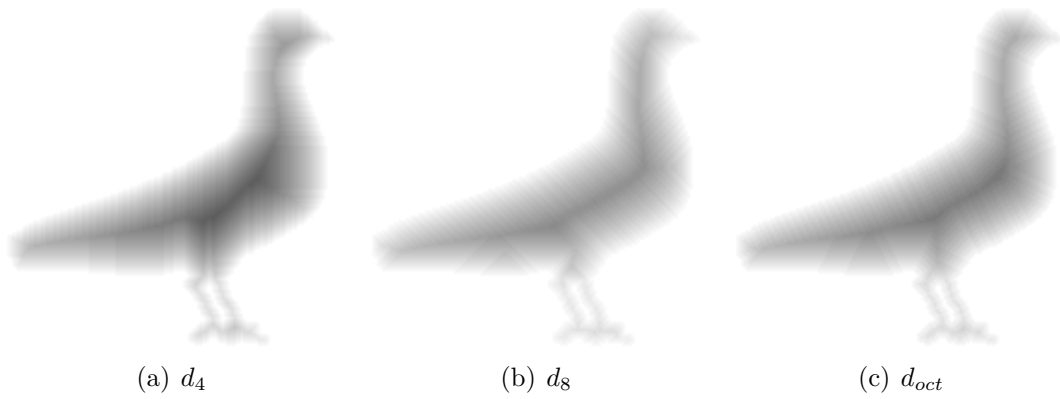


FIGURE 3.14 – Cartes de distance de l'oiseau.

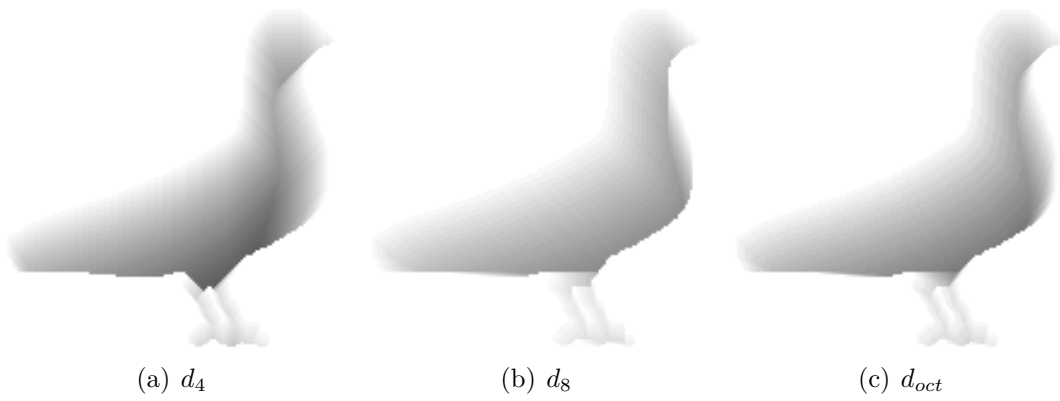


FIGURE 3.15 – Cartes de distance translatées de l'oiseau.

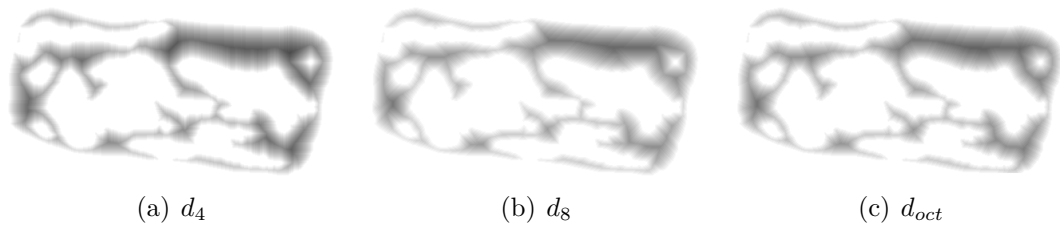


FIGURE 3.16 – Cartes de distance de l'os.



FIGURE 3.18 – Exemple de petit objet.

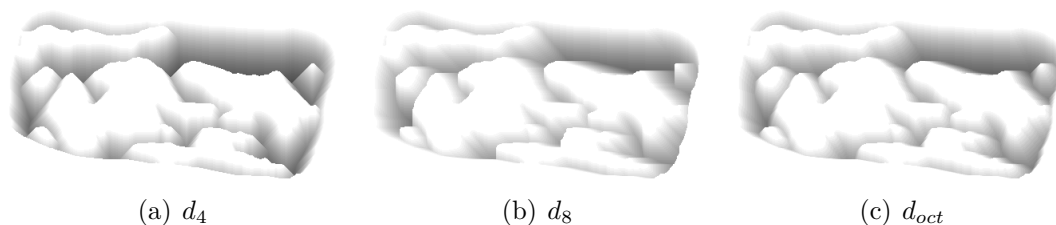


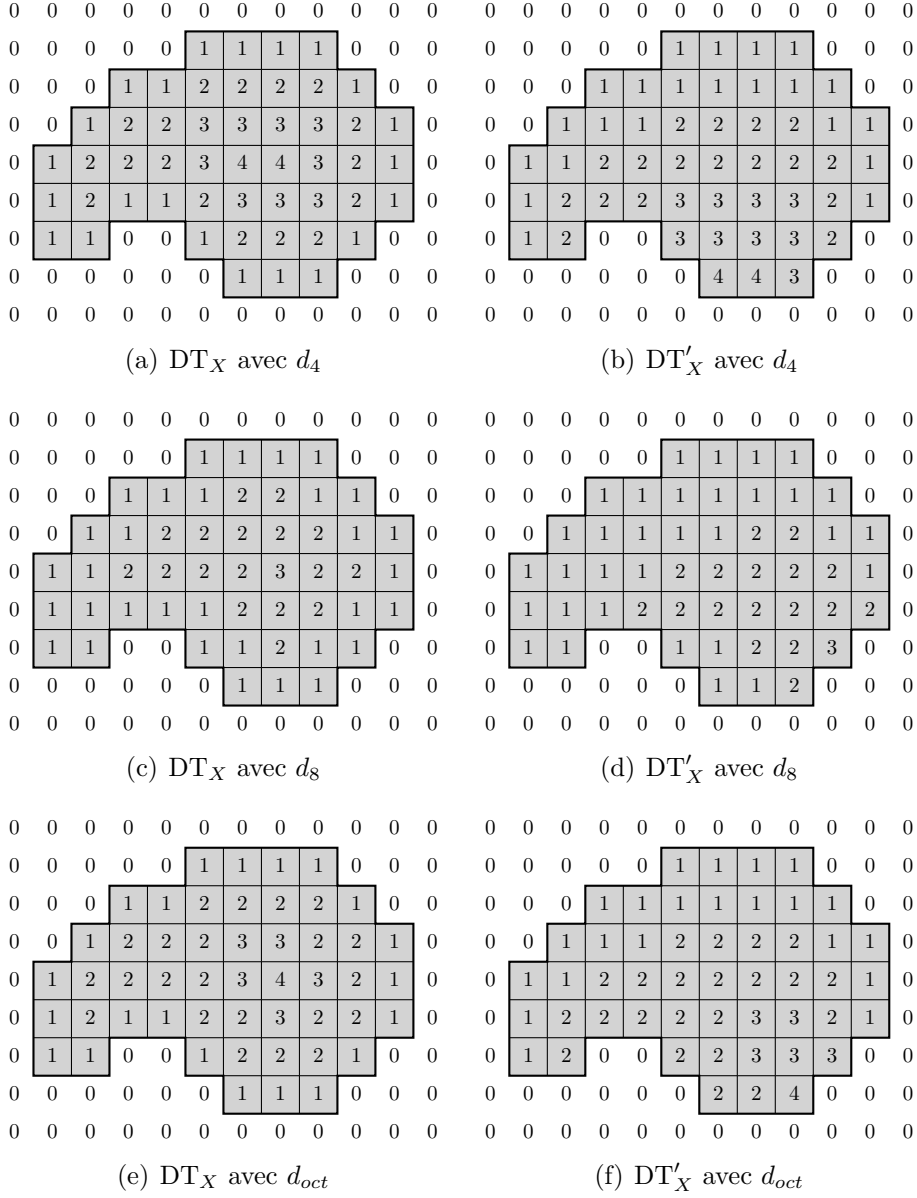
FIGURE 3.17 – Cartes de distance translatées de l'os.

Sur les images des cartes translatées, on peut s'apercevoir que la nuance de gris n'est pas symétriquement répartie dans la forme comme c'est le cas dans les images des cartes de distance translatées. Plus précisément, sur les images des cartes de distance translatées suivant la distance d_4 , on s'aperçoit que la variation du niveau de gris a une direction verticale alors que pour la distance d_8 elle est diagonale. Ceci s'explique par le vecteur de translation $\vec{t}_{(r)}$ utilisé pour traduire les centres des disques.

Pour mieux voir ce phénomène, regardons les valeurs que prennent les cartes (figure 3.19) avec un exemple de petit objet représenté en figure 3.18.

Les cartes de distance d_4 , d_8 et d_{oct} sont respectivement représentées par les figures 3.19(a), 3.19(c) et 3.19(e). Sur ces dernières, l'information distance est symétrique.

Les cartes de distance translatées d_4 , d_8 et d_{oct} sont respectivement représentées par les figures 3.19(b), 3.19(d) et 3.19(f). Ces cartes sont asymétriques.

FIGURE 3.19 – Valeurs de DT_X et DT'_X sur un petit objet.

Les algorithmes 5 (calcul de DT'_X) et 6 (calcul de DT_X à partir de DT'_X) ont été testés successivement sur la base d'images. Ces tests ont montré l'égalité au pixel près de DT_X et de DT_X calculé grâce à l'association des algorithmes 5 et 6.

Afin de se rendre compte de l'efficacité de calcul de l'algorithme 5, nous avons

relevé les temps d'exécution sur la base d'images. Nous avons ensuite comparé ces temps avec ceux relevés avec l'algorithme de référence sur cette même base d'images de test.

Les courbes 3.20, 3.21 et 3.22 montrent respectivement les temps d'exécution de la fonction de distance pour les distances d_4 , d_8 et d_{oct} en fonction du nombre total de pixels de l'image.

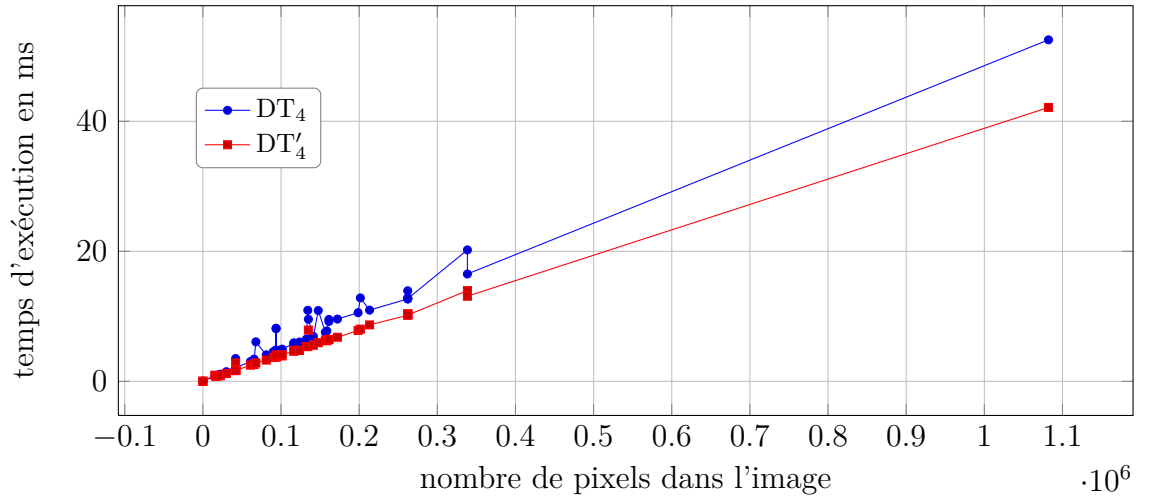


FIGURE 3.20 – Temps de calcul de DT_X et de DT'_X en d_4 en fonction du nombre total de pixels de l'image.

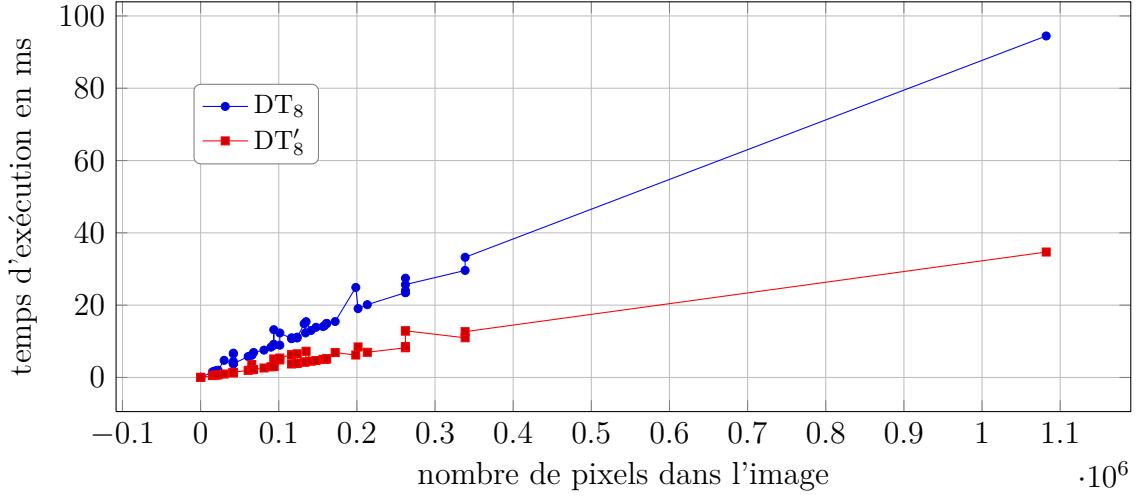


FIGURE 3.21 – Temps de calcul de DT_X et de DT'_X en d_8 en fonction du nombre total de pixels de l'image.

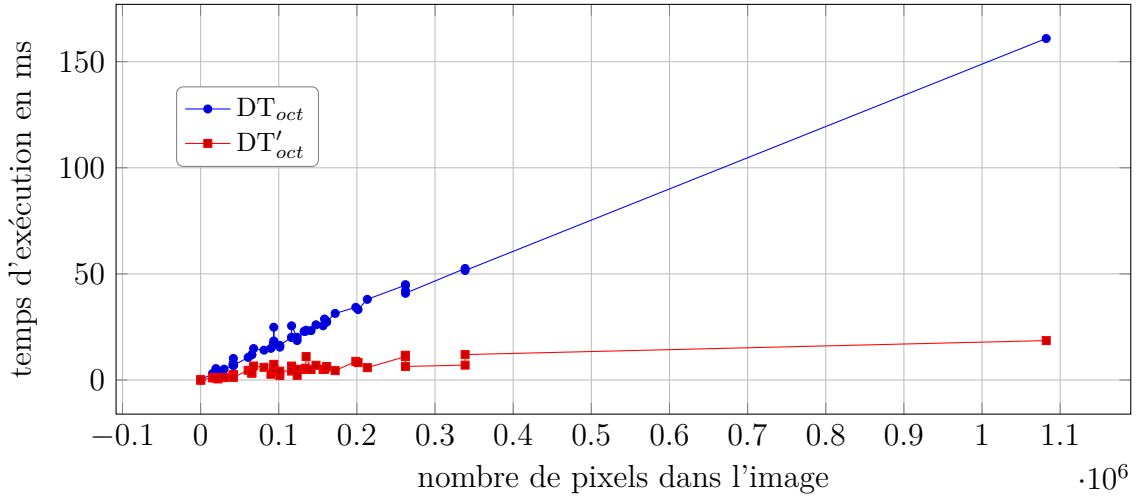


FIGURE 3.22 – Temps de calcul de DT_X et de DT'_X en d_{oct} en fonction du nombre total de pixels de l'image.

Les courbes 3.20, 3.21 et 3.22 montrent également la rapidité de l'algorithme de DT'_X mais elles montrent en plus que cette tendance est accentuée par le nombre de pixels total de l'image. Ceci s'explique par le fait que l'algorithme de DT_X ne procède qu'à un seul balayage de l'image quel que soit le nombre de pixels de l'image, de la forme et quelle que soit la valeur du rayon maximal observé dans

l'image.

Les courbes 7.28, 7.29 et 7.30 montrent respectivement les temps d'exécution de la fonction de distance pour les distances d_4 , d_8 et d_{oct} en fonction du rayon maximal observé dans la forme.

Les courbes 7.25, 7.26 et 7.27 montrent respectivement les temps d'exécution de la fonction de distance pour les distances d_4 , d_8 et d_{oct} en fonction du nombre de pixels de la forme.

En terme de complexité on note deux aspects :

- la complexité temporelle,
- la complexité spatiale.

La complexité temporelle des algorithmes (les nôtres et ceux de la littérature) est linéaire.

Concernant la complexité spatiale, les cartes DT_X et DT'_X sont calculées à la volée. Cependant nos algorithmes pour calculer DT'_X nécessitent deux lignes supplémentaires à partir du point courant. Alors que pour calculer DT_X , il est nécessaire d'avoir un nombre de lignes égal au rayon maximal de la forme moins une ligne.

3.6 Conclusion

Ce chapitre montre comment calculer une carte de distance (DT'_X) en un balayage.

Les algorithmes de la littérature utilisent un ordre de balayage de type “raster scan” en deux passes (ou plus selon la distance utilisée) sur l'image. L'information se propage uniformément de l'extérieur de la forme vers l'intérieur. Notre algorithme se base sur un ordre de balayage de type “raster scan” en une seule passe. De ce fait, l'information se propage en fonction du balayage dans la direction du vecteur de translation des disques utilisés. Cette translation des disques permet un calcul de la carte à la volée où 2 lignes supplémentaires (par rapport au point courant) sont requises alors que le calcul de DT_X nécessite un nombre de lignes égal au rayon maximal de la forme moins une ligne.

Ce changement induit un non-respect de la définition de la distance qui doit

être symétrique. Donc nous ne pouvons pas réellement l'appeler “distance” c’est pourquoi nous parlons toujours de “distance translatée” ou “distance asymétrique”.

Ce chapitre montre également l’équivalence entre une carte de distance et son homologue translatée. C’est-à-dire que grâce à la valeur de distance translatée d’un point ($DT'_X(p')$), son homologue ($DT_X(p)$) est facilement identifiable.

Les temps de calculs de la carte de distance translatée DT'_X sont plus courts que ceux de l’algorithme de référence. Les résultats présentés dans ce chapitre montrent trois exemples de carte de distance. Seules les distances d_4 , d_8 et d_{oct} sont présentées mais l’utilisation d’autres distances est envisageable (d_{ns} avec d’autres séquences par exemple).

Le calcul de DT'_X est la première étape du processus de squelettisation. Dans la seconde étape, décrite dans le chapitre 4, il s’agira de calculer les points de l’axe médian.

Ces points serviront de points d’ancrage au squelette, son calcul sera présenté dans le chapitre 5.

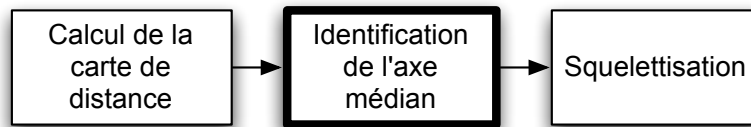
L'axe médian

4.1 Introduction

Dans le chapitre précédent, une carte de distance translatée DT'_X a été calculée. Cette carte est calculée en un balayage de l'image.

La figure 4.1 rappelle les différentes étapes de la squelettisation.

FIGURE 4.1 – Schéma du processus de squelettisation.



Dans ce chapitre est décrite la fonction permettant d'identifier les points de l'axe médian. L'axe médian est l'ensemble des centres des boules maximales incluses dans la forme (définition 10). Rappelons qu'une boule maximale est une boule non incluse dans une autre.

La figure 4.2 illustre un exemple de point de l'axe médian. La valeur 2 encerclée est un point de l'axe médian car aucun autre disque ne le contient entièrement. Le disque de rayon 4 ne recouvre seulement qu'une partie du disque de rayon 2.

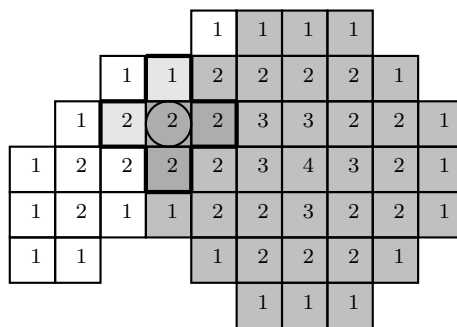


FIGURE 4.2 – Exemple de point de l'axe médian.

L'identification de l'axe médian est la seconde étape pour calculer notre squelette. Il nous servira comme point d'ancrage lors de la squelettisation. Le calcul de l'axe médian est une opération locale. C'est-à-dire qu'il n'y a pas de propagation d'information dans cette étape.

Dans un premier temps, ce chapitre décrit comment obtenir l'axe médian grâce à la carte de distance translatée et dans un second temps nous montrerons les résultats de notre algorithme obtenus avec différentes cartes de distances.

4.2 Calcul de l'axe médian

L'axe médian est l'ensemble des centres des boules maximales complètement incluses dans la forme. Ces points de l'axe médian sont des données intéressantes du point de vue de l'analyse de la forme. Car ces points sont par définition centrés dans la forme et si l'information de distance (rayon de la boule maximale) est sauvegardée, on peut reconstruire la forme à partir de ces points. Cependant l'axe médian ne constitue pas à lui seul le squelette de la forme. Les points de l'axe médian sont des points déconnectés des uns et des autres et il faudra un autre traitement (présenté dans le chapitre 5) pour obtenir le squelette.

Il faut cependant noter que cette manière de calculer les points de l'axe médian n'est pas valable pour toutes les distances (par exemple celles de chanfrein [94]).

Pour les distances que nous utilisons dans ce travail, ce test local est suffisant car d'après Rosenfeld [79] les maxima locaux de la carte de distance sont des points

de l'axe médian.

Un point maximal est un point p qui ne propage pas l'information de distance à ses voisins :

$$\forall \vec{v} \in \mathcal{N}, DT_X(p) \geq DT_X(p + v) \quad (4.1)$$

L'algorithme 7 ci-dessous montre comment calculer les points de l'axe médian. En balayant l'image, on observe les valeurs du centre du disque et de ses voisins. C'est-à-dire que si il n'existe pas de voisin ayant une valeur de distance supérieure au point courant, alors ce point appartient à l'axe médian. On enregistre alors la valeur de la carte de distance aux points de l'axe médian.

Notons également que l'axe médian est réversible dans le sens où il est possible de calculer MA_X à partir de MA'_X . Il suffit pour cela de connaître le vecteur de translation associé au disque maximal et de dé-translater son centre comme dans l'algorithme 6 du calcul de DT_X à partir de DT'_X .

Algorithme 7: Algorithme du calcul de l'axe médian translaté

Données : Carte de distance translatée DT'_X

Résultat : Axe médian MA'_X

```

1 pour  $l \leftarrow 0$  à  $L - 1$  faire
2   pour  $k \leftarrow 0$  à  $K$  faire
3      $r \leftarrow DT'_X(p);$ 
4      $v \leftarrow \max\{DT'_X(p + \vec{v}), \vec{v} \in \mathcal{N}'_{B(r)}\};$ 
5     si  $v \leq r$  alors
6        $MA'_X \leftarrow r;$ 
7     sinon
8        $MA'_X \leftarrow 0;$ 
```

Afin de mieux comprendre cet algorithme prenons l'exemple de la figure 4.3.

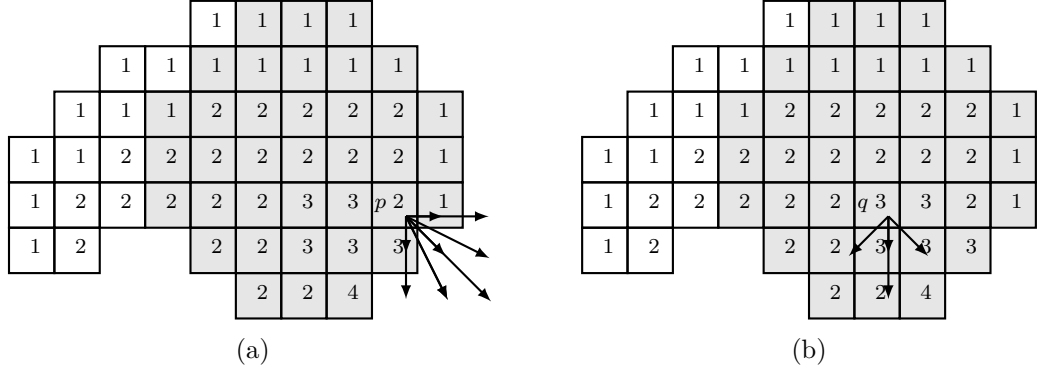


FIGURE 4.3 – Exemple d'identification de l'axe médian sur DT'_{oct} .

Dans l'exemple de la figure 4.3(a), le point p identifié n'est pas un point de l'axe médian. Sa valeur de distance asymétrique est 2 ce qui signifie que, dans l'algorithme et suivant la séquence de d_{oct} , $B(2) = 2$ indique que le voisinage \mathcal{N}'_2 est utilisé pour ce rayon.

Il se trouve qu'un point situé dans \mathcal{N}'_2 du point p a une valeur de distance asymétrique plus grande que r . C'est-à-dire qu'il existe un disque recouvrant le disque de rayon $r = 2$ et de centre p .

Dans le second exemple de la figure 4.3(b), le point q identifié est un point de l'axe médian. $DT'_X(q) = 3$ signifie que l'on va examiner les points situés dans \mathcal{N}'_1 du point q . Dans ce cas toutes les valeurs de distances examinées sont plus petites ou égales à celle du point q , donc le disque de rayon 3 et de centre q n'est pas complètement inclus dans un autre disque ce qui fait de ce point un point de l'axe médian.

Dans le cas où la forme à étudier a une épaisseur paire, les points de l'axe médian sont collés. C'est-à-dire que l'axe médian aura une épaisseur de deux pixels. La figure 4.4 illustre ce cas de figure.

0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	2	2	2	1	0
0	1	2	3	3	3	3	3	3	2	1	0
0	1	2	3	3	3	3	3	3	2	1	0
0	1	2	2	2	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0

(a) MA_X

0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	2	2	2	1	0
0	1	2	2	2	2	2	2	2	2	1	0
0	1	2	2	3	3	3	3	3	3	1	0
0	1	2	2	3	3	3	3	3	3	1	0
0	0	0	0	0	0	0	0	0	0	0	0

(b) MA'_X

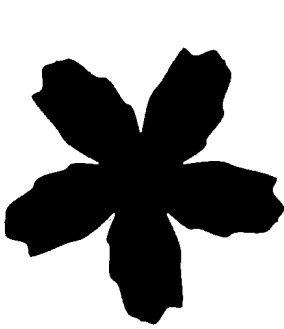
FIGURE 4.4 – Exemple où l'axe médian fait deux pixels d'épaisseur.

La figure 4.4(a) montre les points de l'axe médian (plus foncés) sur une carte de distance d_{oct} classique. En revanche, la figure 4.4(b) montre ces points de l'axe médian sur une carte de distance (d_{oct}) asymétrique.

Cette particularité va être évoquée dans le chapitre de squelettisation suivant car ces points seront considérés comme point d'ancrage dans le processus de suppression de points ce qui impliquera que le squelette obtenu pourra ne pas être fin.

4.3 Résultats

Dans cette section sont présentés les résultats de l'extraction de l'axe médian. La figure 4.5 présente les 3 formes binaires que nous allons étudier.



(a) Fleur



(b) Oiseau



(c) Os

FIGURE 4.5 – Images originales binaires.

Les figures 4.6, 4.8 et 4.10 présentent les axes médians des 3 formes calculés avec les distances d_4 , d_8 et d_{oct} . Tandis que les figures 4.7, 4.9 et 4.11 présentent les axes médians translatés calculés à partir des cartes de distances asymétriques de d_4 , d_8 et d_{oct} .

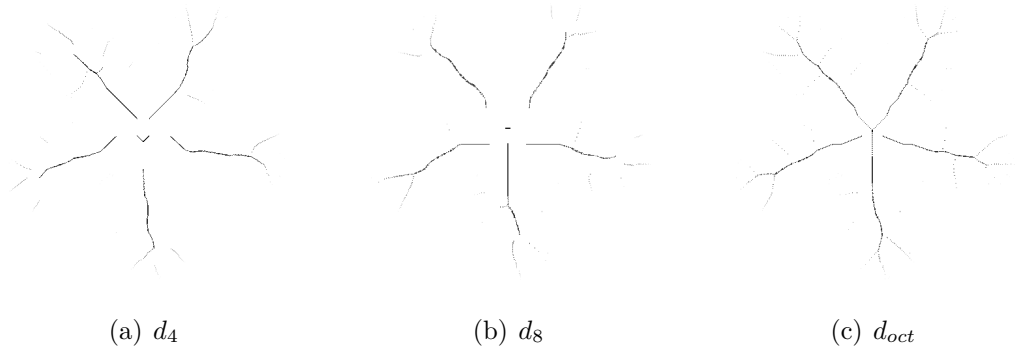


FIGURE 4.6 – Axe médian de la fleur.

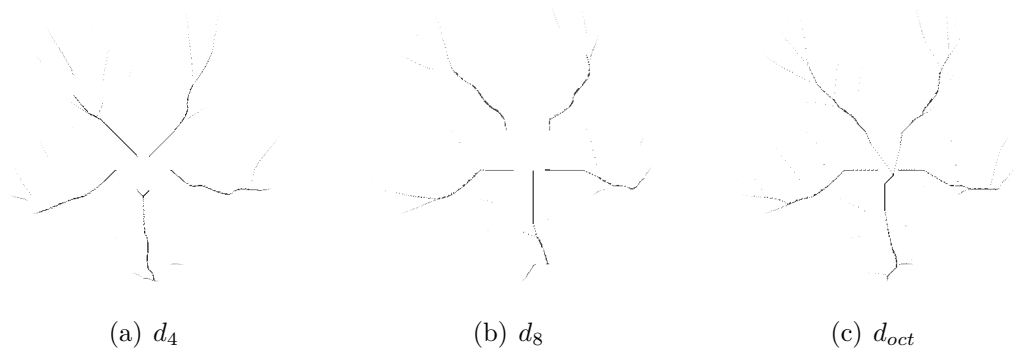


FIGURE 4.7 – Axe médian translaté de la fleur.

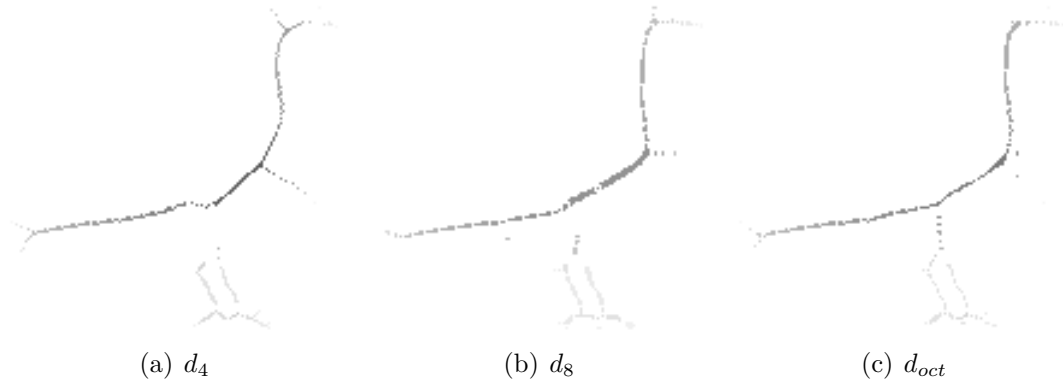


FIGURE 4.8 – Axe médian de l'oiseau.

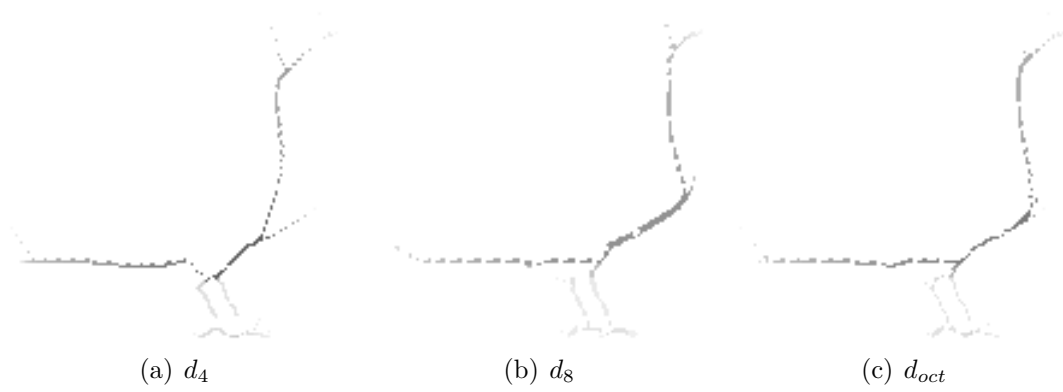


FIGURE 4.9 – Axe médian translaté de l'oiseau.

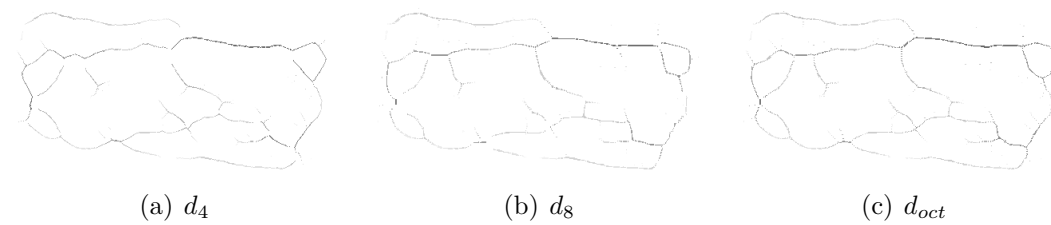


FIGURE 4.10 – Axe médian de l'os.

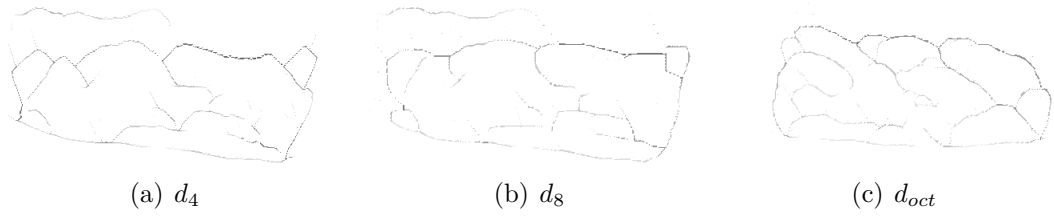


FIGURE 4.11 – Axe médian translaté de l'os.

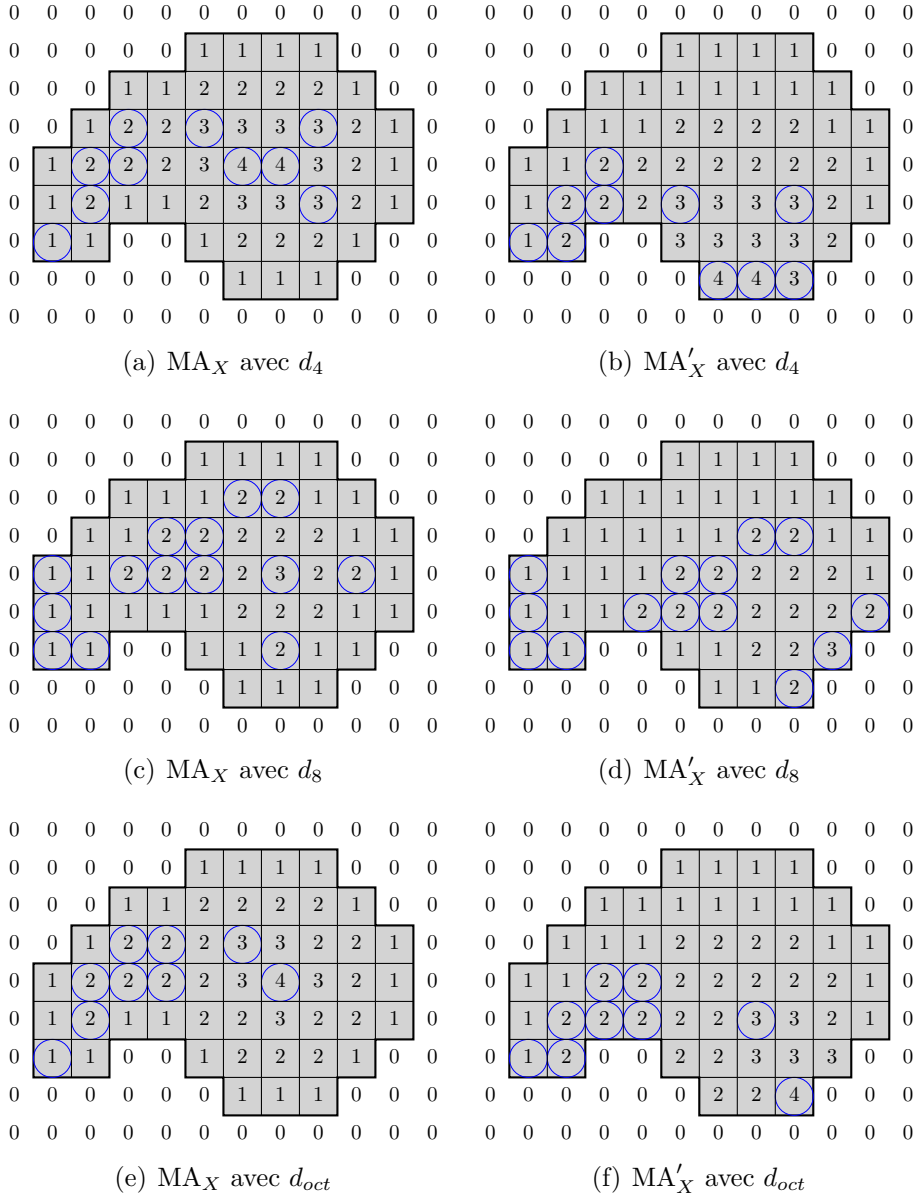
On remarque qu'en comparaison avec les points de l'axe médian non translatés, les points de l'axe médian translatés sont légèrement translatés. Cette translation observée est en accord avec le vecteur \vec{t} utilisé pour translater les centres des disques.

La figure 4.12 présente une forme binaire que nous allons étudier pixel par pixel.



FIGURE 4.12 – Exemple de petit objet.

La figure 4.13 présente les cartes de distances et les points de l'axe médian associés à l'image de la figure 4.12.

FIGURE 4.13 – Exemples d’axes médians calculés à partir de DT_X et DT'_X .

On remarque que l’axe médian calculé à partir de la carte de distance tradatée d_4 se trouve “écrasé” en bas de l’image. Tandis que l’axe médian calculé à partir de la carte de distance tradatée d_8 se trouve “écrasé” en bas à droite de l’image, selon le vecteur de translation \vec{t} . Cette particularité est due au sens de translation du voisinage de la boule.

L'identification de l'axe médian sur une carte de distance DT_X ou DT'_X se déroule de la même manière : une suite d'opérations locales sur des voisinages de la même taille est appliquée. De ce fait, les temps d'exécution des algorithmes basés sur les cartes de distances translatées et non translatées sont les mêmes. Les figures 7.34 à 7.36 présentent respectivement les courbes des temps d'exécution pour les algorithmes d'extraction de l'axe médian sur les cartes de distance DT_X et DT'_X en d_4 , d_8 et d_{oct} .

Ces temps d'exécution ont été relevés sur 57 images de différentes tailles calculées chacune 100 fois, les temps indiqués sont donc une moyenne de ces 100 relevés.

Ces figures nous indiquent deux choses :

- il n'y a pas de réelle différence de temps d'exécution entre nos algorithmes et les algorithmes de référence,
- l'allure des trois courbes montre que le temps d'exécution du calcul de l'axe médian ne dépend pas de la distance utilisée.

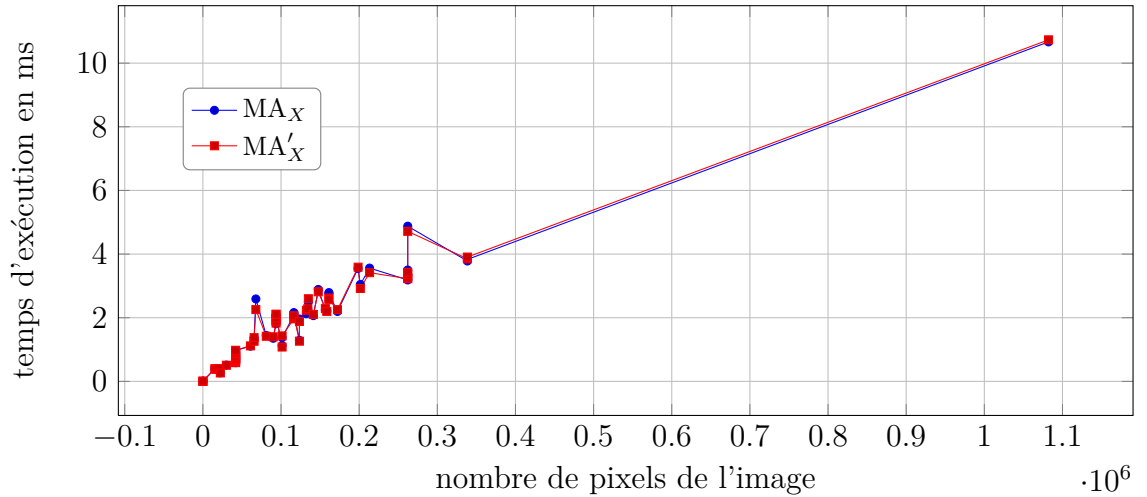


FIGURE 4.14 – Temps de calcul de MA_X et de MA'_X en d_4 en fonction du nombre de pixels de l'image.

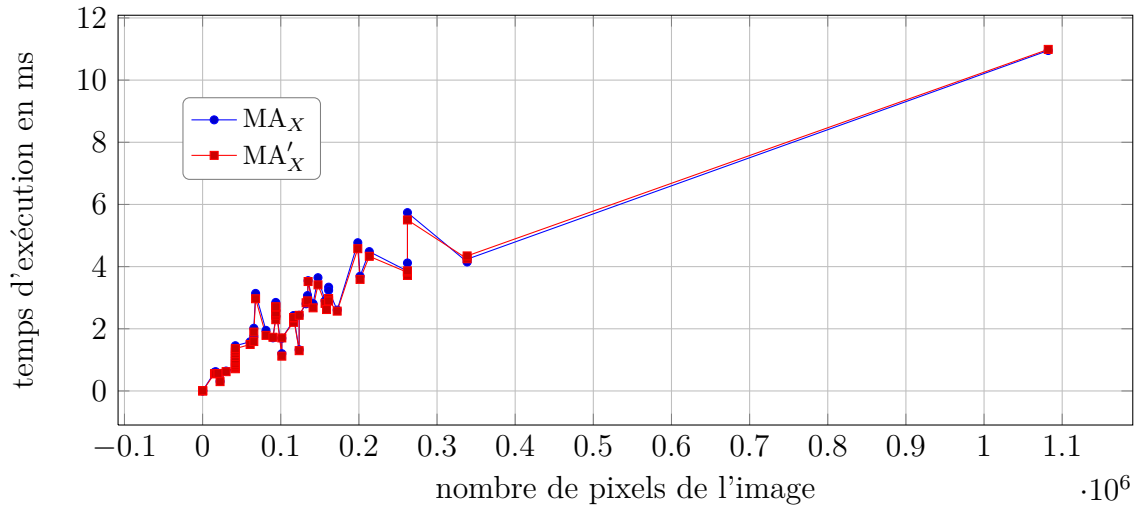


FIGURE 4.15 – Temps de calcul de MA_X et de MA'_X en d_8 en fonction du nombre de pixels de l'image.

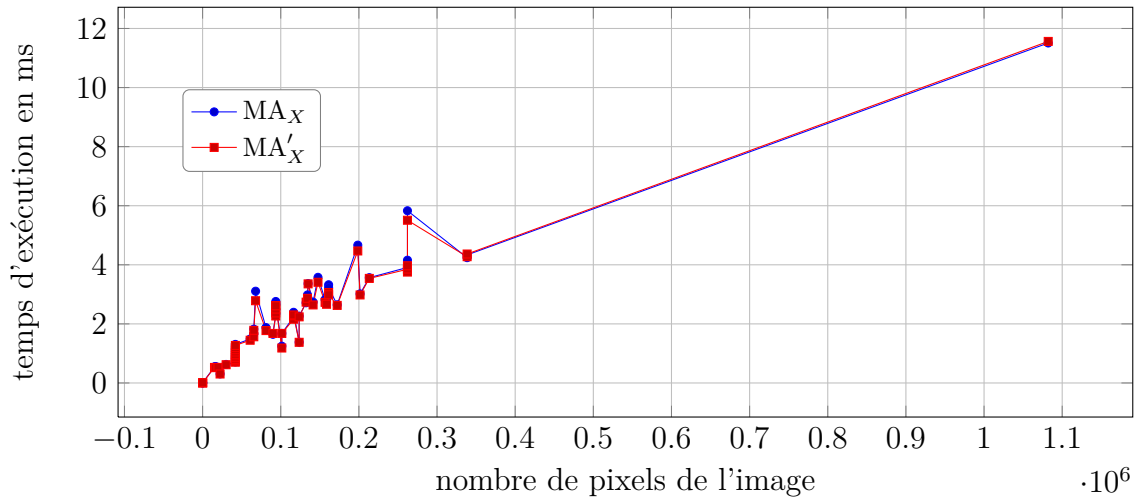


FIGURE 4.16 – Temps de calcul de MA_X et de MA'_X en d_{oct} en fonction du nombre de pixels de l'image.

4.4 Conclusion

Dans ce chapitre a été présenté le calcul de l'axe médian MA'_X en un balayage de l'image.

L'identification des points de l'axe médian est la deuxième étape dans le processus de squelettisation choisi. Ce calcul n'induit pas de propagation de l'information car c'est un calcul local. C'est pourquoi les temps d'exécutions sont à peu près équivalents aux temps de calcul de l'algorithme de référence. Ces points serviront de points d'ancrage dans le processus de squelettisation dont le processus est présenté dans le chapitre 5 suivant.

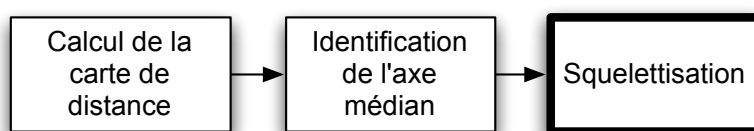
On note également que lorsque la forme a une épaisseur paire, l'axe médian a une épaisseur de deux pixels ce qui impliquera que le squelette pourra également avoir une épaisseur de deux pixels. Nous attirons l'attention sur le fait que les points de l'axe médian sont déconnectés les uns des autres. C'est pourquoi on ne peut pas parler de squelette à cette étape.

La squelettisation

5.1 Introduction

Ce chapitre décrit le calcul du squelette à partir de la carte de distance translée et de l'axe médian translé. La figure 5.1 rappelle les différentes étapes du processus de squelettisation.

FIGURE 5.1 – Schéma du processus de squelettisation.



La particularité de notre algorithme de squelettisation résulte dans le fait que le squelette est calculé avec un balayage unique de l'image.

Dans son travail de thèse, Nicolas Normand [57] extrait des squelettes sur une carte de distance “asymétrique”. Pour cela, il identifie les points de l'axe médian puis des points selles. Après cette phase d'identification des points d'ancrage, il les reconnecte en suivant les gradients de l'image. Cette méthode ne permet pas de garantir l'homotopie du squelette.

Dans ce travail, nous allons présenter une autre méthode permettant d'extraire le squelette.

Dans ce chapitre est décrite cette méthode de squelettisation calculée en un balayage en insistant sur deux aspects importants quant aux résultats :

- le choix des points d'ancrage,
- l'ordre dans lequel est appliquée la suppression des points simples.

Dans ce travail, un algorithme de référence a été choisi selon l'ordonnancement de ses instructions. Plus précisément, selon la dépendance des pixels calculés dans les boucles d'instructions.

On notera donc trois boucles principales dans l'algorithme de référence retenu : la première sur les lignes de l'image, la seconde sur les colonnes et la troisième sur les rayons de la carte de distance. Entre l'algorithme de référence et l'algorithme de squelettisation en un balayage, se situe un algorithme intermédiaire. Cet algorithme intermédiaire permet de transformer progressivement l'algorithme de référence en l'algorithme de squelettisation en un balayage en changeant l'ordre des boucles principales. Dans ce chapitre sont décrits les différents algorithmes : référence (appelé *rlk* dans la section 5.2), intermédiaire (appelé *l'rk'* dans la sous-section 5.3.1) et l'algorithme en un balayage (appelé *l'k'r* dans la sous section 5.3.2).

5.2 Algorithme de référence

Le squelette calculé est le résultat d'un amincissement contrôlé par la distance (squelettisation hybride présentée dans la section 2.5). Comme vu précédemment, l'amincissement garantit l'homotopie et le contrôle par la carte de distance permet d'obtenir un résultat centré puisque les points de l'axe médian sont des points d'ancrage du squelette qu'on ne peut pas supprimer. Ce procédé ne permettra pas d'obtenir un squelette fin. L'algorithme d'amincissement contrôlé par la distance permet de calculer un squelette, par le biais de plusieurs balayages de l'image, de manière séquentielle.

Soit r les valeurs de distance de la carte de distance associée à l'image binaire X . La méthode d'amincissement consiste à supprimer les points simples du contour. Cette méthode supprime les points de $r = 1$ à r_{\max} . Les algorithmes de squelettisation ordonnée par la distance nécessitent r_{\max} balayages de l'image pour calculer le squelette. Ceci est vrai sauf pour certaines méthodes comme l'algorithme de squelettisation *DOHT* [71] car il ne fonctionne pas vraiment sur des principes de balayages de l'image mais avec une file *FIFO*.

De manière générale, les méthodes séquentielles de squelettisation ordonnée par la distance [90] parcourent l'image suivant, dans un premier temps, les rayons par

ordre croissant, puis les lignes et enfin les colonnes de l'image.

Pour chaque rayon et pour chaque point une fonction de simplicité est appliquée. La fonction de simplicité utilisée est une application du calcul du “*crossing number*” d'Hilditch (introduite dans la sous section 1.3.3) :

$$X_H(p) = \sum_{i=1}^4 b_i \quad (5.1)$$

$$\text{où } b_i = \begin{cases} 1 & \text{si } x_{2i-1} = 0 \text{ et } (x_{2i} = 1 \text{ ou } x_{2i+1} = 1) \\ 0 & \text{sinon} \end{cases}$$

L'algorithme 8 est une proposition de vérification de cette simplicité. Les points du voisinage sont identifiés selon la figure 5.2.

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

FIGURE 5.2 – Représentation du voisinage du point p .

Algorithme 8: Algorithme de simplicité.

Données : Image binaire X , point p

Résultat : 0 ou 1

```

1 si ( $\bar{x}_1$  ET ( $x_2$  OU  $x_3$ )
2 +  $\bar{x}_3$  ET ( $x_4$  OU  $x_5$ )
3 +  $\bar{x}_5$  ET ( $x_6$  OU  $x_7$ )
4 +  $\bar{x}_7$  ET ( $x_8$  OU  $x_1$ )) = 1 alors
5   └─ retourne 1;
```

Si le point est simple, il ne fait pas partie du squelette. En revanche si le point n'est pas simple ou s'il fait partie de l'axe médian alors il fait partie des points du squelette.

Nous faisons l'hypothèse que pour une telle méthode, les points sont examinés une seule fois l'ordre croissant de leur distance r et dans l'ordre d'apparition du point dans l'image. Pour valider cette hypothèse, nous allons vérifier l'incidence

de l'ordre de balayage sur la dépendance des pixels.

La suppression d'un point peut entraîner une propagation de la simplicité dans son voisinage. La question à laquelle répond la relation d'ordre partiel peut être formulée de cette façon : “si on supprime un point simple, quels sont les points qui sont potentiellement affectés, *i.e.* dont il faut tester la simplicité après?”.

Il est possible de mettre en œuvre différentes versions d'algorithmes séquentiels. Chacune traite les pixels dans un ordre, dit “complet”, particulier. Cet ordre peut être compatible (ce qui est appelé tri topologique en théorie des graphes) ou non avec l'ordre partiel.

La relation de dépendance locale est définie comme suit : soient deux points p et $q \in X$, de coordonnées respectives (k_p, l_p) et (k_q, l_q) , tels que $q \in \mathcal{N}_2(p)$ alors :

$$p \prec q \iff \begin{cases} DT_X(p) < DT_X(q) \\ ou \\ DT_X(p) = DT_X(q) \text{ et } l_p < l_q \\ ou \\ DT_X(p) = DT_X(q) \text{ et } l_p = l_q \text{ et } k_p < k_q \end{cases} \quad (5.2)$$

Lorsque p et q ne sont pas voisins, il ne sont pas directement comparables. Cependant, ils peuvent être comparables par transitivité de la relation.

Nous allons présenter dans ce travail trois algorithmes dont l'ordre de traitement séquentiel des pixels (qui est forcément complet) est différent. Afin de garantir l'identité des résultats au pixel près, nous allons vérifier que ces trois ordres complets respectent la relation de dépendance (ordre partiel) définie précédemment dans l'équation 5.2.

Le respect de l'ordre partiel dans les ordres complets présentés ci-après assure l'identité des résultats au pixel près.

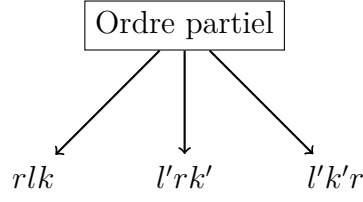


FIGURE 5.3 – Ordre partiel et ordres complets.

L'algorithme 9 *rlk* possède trois boucles d'instructions :

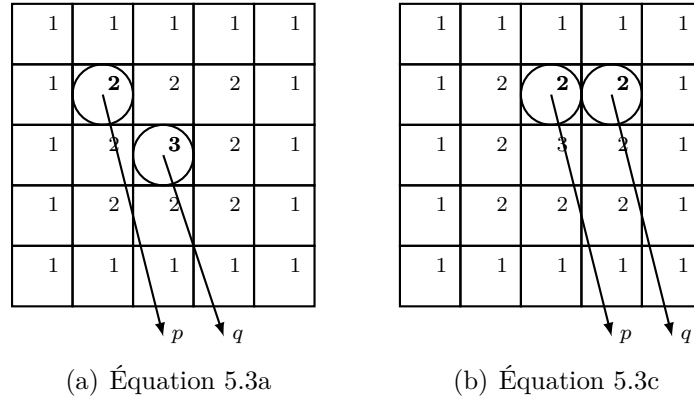
- la boucle principale parcourt les valeurs des rayons r de DT_X , elle implique l'équation Eq. (5.3a),
- la seconde boucle parcourt les lignes l et implique l'équation Eq. (5.3b),
- la dernière boucle parcourt les colonnes k et implique l'équation Eq. (5.3c).

L'ordre de calcul des points dans l'algorithme *rlk* suit naturellement l'ordre partiel :

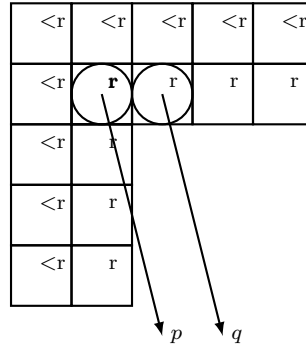
$$p \prec^{rlk} q \iff \begin{cases} DT_X(p) < DT_X(q) & (5.3a) \\ \text{ou} \\ DT_X(p) = DT_X(q) \text{ et } l_p < l_q & (5.3b) \\ \text{ou} \\ DT_X(p) = DT_X(q) \text{ et } l_p = l_q \text{ et } k_p < k_q & (5.3c) \end{cases}$$

avec $p \prec^{rlk} q$ qui veut dire que le point p est traité avant le point q dans l'ordre *rlk*.

L'ordre de calcul de l'algorithme *rlk* respecte la relation de dépendance des pixels. La figure 5.4 illustre l'ordre dans lequel sont traités les pixels. Dans la figure 5.4(a) le point p a une valeur de distance inférieure à celle du point q . De ce fait, p sera calculé avant q . En revanche, dans la figure 5.4(b) les points p et q ont la même valeur de distance et sont sur la même ligne. Dans ce cas le point ayant la plus petite valeur de k (colonne) sera calculé avant l'autre point, ici p sera calculé avant q .


 FIGURE 5.4 – Exemple de deux cas où $p \not\prec^{rlk} q$.

L'hypothèse émise selon laquelle chaque pixel de la forme est examiné une seule fois est vérifiée, à l'exception d'un seul cas. Ce cas se produit lorsque la forme présente une concavité dont l'angle la formant est orienté dans le sens inverse du balayage comme l'illustre la figure 5.5.


 FIGURE 5.5 – Cas où la suppression entraîne une modification de la topologie de la forme si l'on supprime le point p (création d'un trou).

Dans ce cas, lors du balayage sur les valeurs de distance r , le premier point examiné est p . Ce point n'est pas simple puisque sa suppression entraînerait la création d'un trou. À l'itération suivante, son voisin q est simple donc sera supprimé. La suppression du point q entraîne la simplicité du point p . Dans ce cas uniquement, lors de la suppression du point q nous re-testons la simplicité du point p .

On note que ce cas ne survient pas lors de l'utilisation de la distance d_4 , ou plus

généralement dans le voisinage \mathcal{N}_1 du point, puisqu'à ce point "critique" (p) la distance est $r + 1$ donc ce point est examiné dans le balayage suivant.

Algorithme 9: Algorithme rayon/ligne/colonnes (rlk)

Données : Image binaire X , DT_X , MA_X

Résultat : Le squelette Sk

```

1 pour  $r \leftarrow 1$  à  $\max_{p \in X} \{DT_X(p)\}$  faire
2   pour  $l \leftarrow 0$  à  $L - 1$  faire
3     pour  $k \leftarrow 0$  à  $K - 1$  faire
4       si  $DT_X(k, l) = r$  et  $MA_X(k, l) \neq r$  et
5          $simple(k, l)$  alors
6            $Sk_X(k, l) \leftarrow 0;$ 

```

5.3 De rlk vers $l'k'r$

L'algorithme rlk , qui balaye l'image par ordre croissant des rayons et supprime les points simples, est notre algorithme de référence. Cet algorithme procède à r_{\max} (rayon maximal de l'image) balayages de l'image pour calculer le squelette. Cette section présente la manière dont l'ordre de balayage de l'image a été modifié pour obtenir un algorithme en un seul balayage appelé $l'k'r$ tout en respectant l'ordre partiel précédemment défini.

5.3.1 Algorithme ordonné en ligne/rayon de disques/colonne ($l'rk'$)

Dans la suite de ce travail on notera p et q , respectivement de coordonnées (k_p, l_p) et (k_q, l_q) , les points dans la carte de distance non translatée. De la même manière, on notera p' et q' , respectivement de coordonnées (k'_p, l'_p) et (k'_q, l'_q) , les points dans la carte de distance translatée. Donc on a $p' = p + \vec{t}_r$ pour un disque de rayon r .

La translation de vecteur correspondant à la distance d_4 est $\overrightarrow{t_1(r)} : (0, r - 1)$ et la translation de vecteur correspondant à la distance d_8 est $\overrightarrow{t_2(r)} : (r - 1, r - 1)$.

Plus g n ralement, soit une s quence de voisinage $B(r)$:

$$\overrightarrow{t_B(r)} = \overrightarrow{t_B(r-1)} + \overrightarrow{t_{B(r)}}. \quad (5.4)$$

o  $\overrightarrow{t_{B(r)}}$ correspond au vecteur de translation de la derni re s quence de voisinage utilis e :

$$\overrightarrow{t_{B(r)}} : (\mathbf{2}_B(r), \mathbf{1}_B(r) + \mathbf{2}_B(r)) = (\mathbf{2}_B(r), r). \quad (5.5)$$

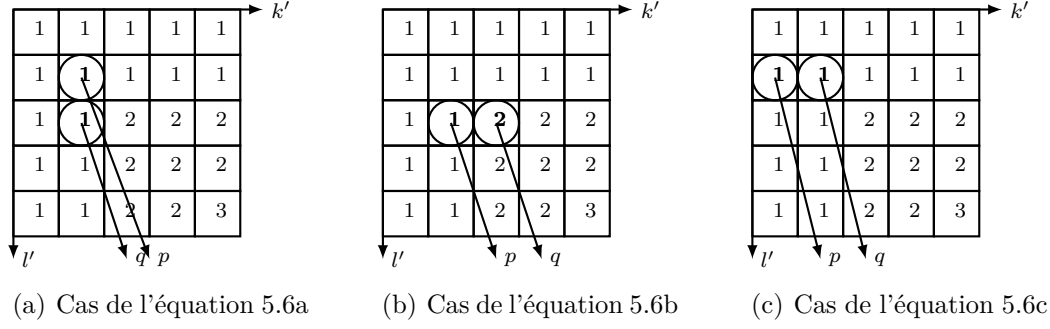
Proposition 1. *Si l'on change le sens de parcours de l'image en commen ant par examiner les lignes puis les rayons puis les colonnes de l'image on obtient l'ordre complet de l'algorithme $l'rk'$ suivant :*

$$p \prec^{l'rk'} q \iff \begin{cases} l'_p < l'_q & (5.6a) \\ \text{ou} \\ l'_p = l'_q \text{ et } DT'_X(p') < DT'_X(q') & (5.6b) \\ \text{ou} \\ l'_p = l'_q \text{ et } DT'_X(p') = DT'_X(q') \text{ et } k'_p < k'_q & (5.6c) \end{cases}$$

O  p se trouve avant q quand la ligne de p est inf rieure   celle de q (exemple figure 5.6(a)).

Dans le cas o  les lignes de p et q sont  gales, si la valeur dans la carte de distance au point p est inf rieure   celle de q alors p se trouve avant q (exemple figure 5.6(b)).

D'autre part, si les valeurs de la carte de distance aux deux points sont  gales et que la colonne de p se trouve avant celle de q , alors le point p se trouve avant q dans l'ordre de parcours appel  $l'rk'$ (exemple figure 5.6(c)).


 FIGURE 5.6 – Exemple des cas où $p \prec^{l'rk'} q$.

Le changement du sens de parcours de l'image entraîne une modification de la dépendance entre pixels et un non respect de l'ordre partiel. Il faut alors modifier l'ordre complet $l'rk'$ pour que l'ordre partiel soit respecté et donc que le résultat soit identique à celui de l'algorithme de référence :

$$l'_p = l_p + DT_X(p) \quad (5.7)$$

$$p \prec q \iff \begin{cases} DT_X(p) < DT_X(q) \text{ et } l_p \leq l_q + 1 \Rightarrow DT_X(p) < DT_X(q) \text{ et } l'_p < l'_{q+1} \\ DT_X(p) = DT_X(q) \text{ et } p \prec^{rlk} q \Rightarrow p \prec^{l'rk'} q \end{cases} \quad (5.8)$$

$$l'_p < l'_{q+1} \iff \begin{cases} l'_p < l'_q \Rightarrow p \prec^{l'rk'} q \\ l'_p = l'_q \Rightarrow p \prec^{l'rk'} q \end{cases} \quad (5.9)$$

$$l'_p < l'_{q+1} \iff \begin{cases} l'_p < l'_q \iff l_p + DT_X(p) < l_q + DT_X(q) \\ l'_p = l'_q \iff l_p + DT_X(p) = l_q + DT_X(q) \end{cases} \quad (5.10)$$

Grâce aux équations 5.8 et 5.9 illustrées dans les figures 5.7 et 5.8, on se rend compte que l'algorithme $l'rk'$ traite les points dans un ordre qui respecte leur relation de dépendance.

À partir des équations Eq. (5.6) et Eq. (5.8) $p \prec^{rlk} q \iff p \prec^{l'rk'} q$ nous pouvons

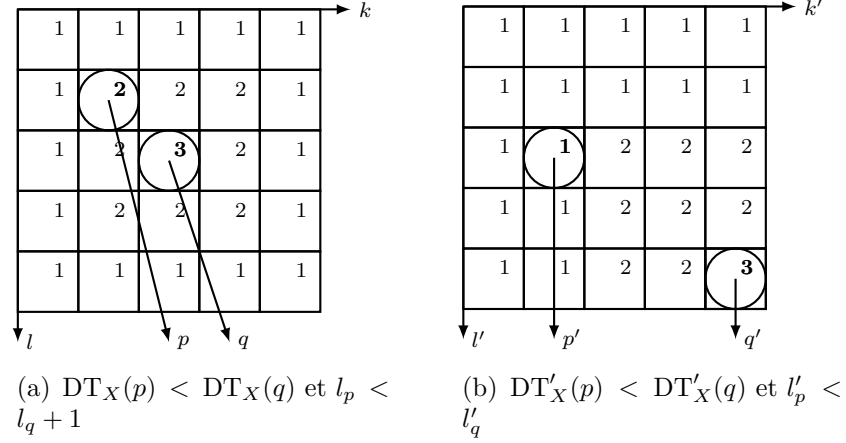


FIGURE 5.7 – Équivalence de la relation de dépendance partielle et de la relation ordonnée par $l'rk'$.

ainsi utiliser l'algorithme 10.

Algorithme 10: Algorithme ligne'/rayon/colonnes' ($l'rk'$)

Données : Une image binaire X , DT'_X , MA'_X

Résultat : Le squelette Sk

```

1  pour  $l' \leftarrow 0$  à  $L - 1$  faire
2      pour  $r \leftarrow 1$  à  $\max_{p \in X} \{DT'_X(p)\}$  faire
3          pour  $k' \leftarrow 0$  à  $K - 1$  faire
4               $p' \leftarrow (k', l')$ ;
5               $p \leftarrow p' - \vec{t}_r$ ;
6              si  $DT'_X(p + \vec{t}_{r+1}) \leq r$  et  $DT'_X(p') \geq r$  et  $MA'_X(p') \neq r$  et
               simple( $p$ ) alors
7                   $Sk_X(p) \leftarrow 0$ ;
```

5.3.2 Algorithme ordonné en ligne/colonne/rayon de disques ($l'k'r$)

Comme la précédente manipulation, le sens de balayage de l'image a changé afin d'examiner les lignes suivies des colonnes puis des rayons de disques. Cette configuration permet la squelettisation en un balayage unique de l'image.

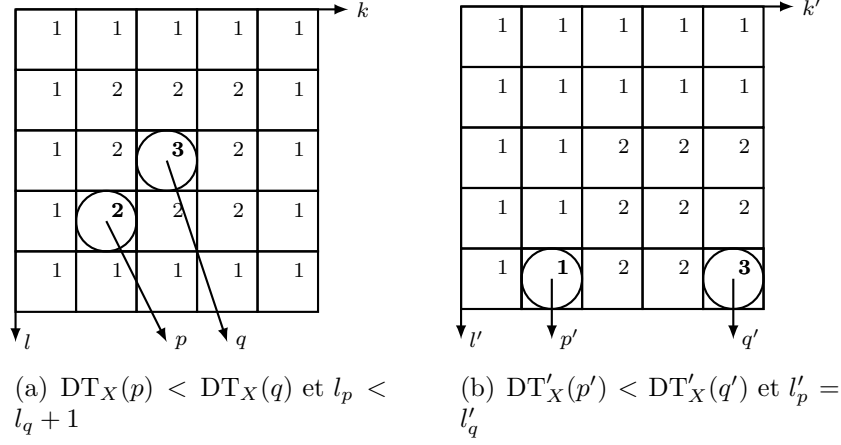


FIGURE 5.8 – Équivalence de la relation de dépendance partielle et de la relation ordonnée par $l'rk'$.

L'ordre total de l'algorithme $l'k'r$ est le suivant :

$$p \prec^{l'kr'} q \iff \begin{cases} l'_p < l'_q \text{ (voir Eq. (5.8))} \\ \text{ou} \\ l'_p = l'_q \text{ et } k'_p < k'_q \\ \text{ou} \\ l'_p = l'_q \text{ et } k'_p = k'_q \text{ et } DT'_X(p') < DT'_X(q') \end{cases} \quad \begin{matrix} (5.11a) \\ (5.11b) \\ (5.11c) \end{matrix}$$

Cet ordre total respecte l'ordre partiel si :

$$p \prec^{l'kr'} q \text{ et } p \in \mathcal{N}q \Rightarrow p \prec q$$

$$l'_p < l'_q \text{ dans 3 cas : } \begin{cases} DT'_X(p) < DT'_X(q) : \text{ si } p \text{ et } q \text{ sont comparables} \\ DT'_X(p) = DT'_X(q) : \text{ on a aussi } l_p < l_q \\ DT'_X(p) > DT'_X(q) : p \text{ et } q \text{ ne sont pas comparables} \end{cases} \quad (5.12)$$

Soit $q \in \mathcal{N}2(p)$, nous savons déjà que $k_p = k_q + i : i \in \{-1, 0, 1\}$ pour les deux types de voisinage ($\mathcal{N}1$ et $\mathcal{N}2$). D'après le vecteur de translation \vec{t}_r on peut savoir

que

$$l'_p = l'_q \text{ et } k'_p < k'_q \iff k_p = k_q - 1 \text{ et } l_p + \text{DT}_X(p) = l_q + \text{DT}_X(q)$$

Avec $q \in \mathcal{N}2(p)$, $l_p + \text{DT}_X(p) = l_q + \text{DT}_X(q)$ et $\text{DT}_X(p) > \text{DT}_X(q)$ conduit à $l_p = l_q - 1$ et $\text{DT}_X(p) = \text{DT}_X(q) + 1$.

De la même façon, $\text{DT}_X(p) = \text{DT}_X(q)$ et $\begin{cases} l_p > l_q \\ l_p = l_q \text{ et } k_p > k_q \end{cases}$ n'est pas possible.

Afin de comprendre comment la dépendance entre pixels est respectée dans les ordres $\prec^{l'k'r}$ et \prec^{rlk} , nous allons décomposer l'équation 5.11 pour les différents types de voisinages $\mathcal{N}1$ et $\mathcal{N}2$.

Cas de figure pour le voisinage $\mathcal{N}2$ Dans le cas du voisinage $\mathcal{N}2$ (comme c'est le cas pour la distance d_8), la décomposition de l'équation 5.11 est la suivante :

La première partie de l'équation 5.11a consiste à dire que si la ligne translatée du point p est inférieure à celle de q alors p est calculé avant q comme illustré dans la figure 5.9.

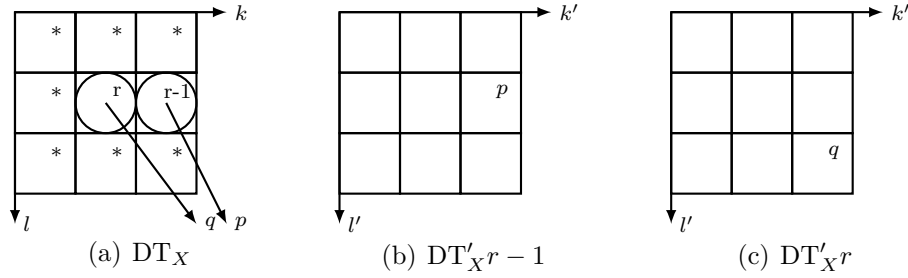


FIGURE 5.9 – Cas de l'équation 5.11a.

La seconde partie de l'équation 5.11b montre que si les lignes traduites des points p et q sont égales, alors l'ordre des colonnes définit l'ordre des points à calculer. Un exemple de configuration est illustré dans la figure 5.10. L'ordre de calcul des points reste cohérent avec l'ordre de calcul dans l'algorithme de référence (équation 5.3a).

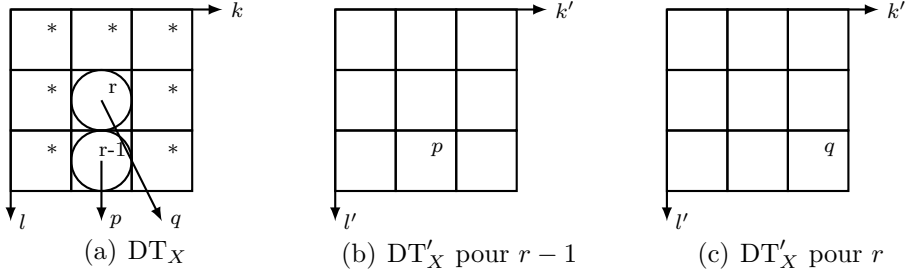


FIGURE 5.10 – Cas de l'équation 5.11b.

La dernière partie de l'équation 5.11c montre que les points p et q peuvent être situés à la même ligne et la même colonne. Dans ce cas, l'algorithme dépile les points dans l'ordre croissant des valeurs de rayons. La figure 5.11(a) montre ce cas de figure. Lorsque que l'on translate les centres des disques (figure 5.11(d)) par le biais du vecteur de translation \vec{t}_r , les trois centres des disques correspondant à p, q et z se trouvent à la même ligne et à la même colonne. Donc z sera traité avant p qui sera traité avant q , comme c'est le cas dans l'algorithme de référence.

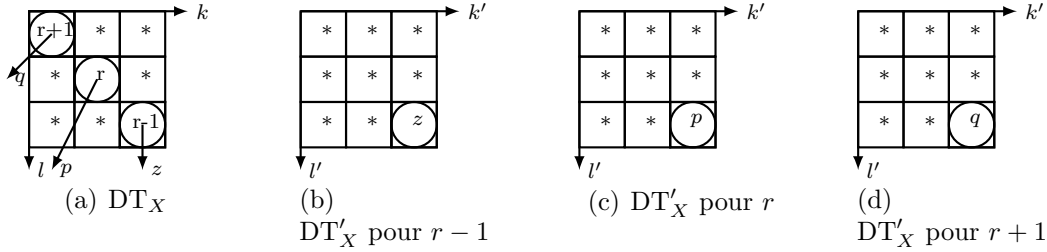


FIGURE 5.11 – Cas de l'équation 5.11c.

Cas de figure pour le voisinage $\mathcal{N}1$ La figure 5.12 illustre un cas de figure où le point p est calculé avant le point q dans l'ordre $l'k'r$ quand la ligne translatée de p est inférieure à celle de q . Dans ce cas de figure, l'algorithme de référence calcule également le point p avant le point q car $DT_X(p) < DT_X(q)$ (voir l'équation 5.11c).

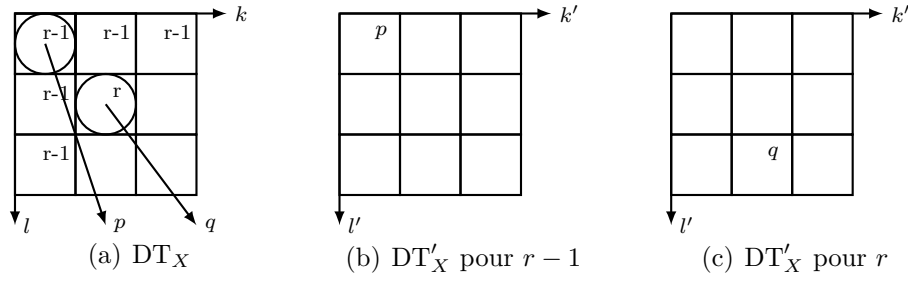


FIGURE 5.12 – Cas de l'équation 5.11a

La seconde partie de l'équation 5.11b explique que si les lignes traduites des points p et q sont égales, c'est l'ordre des colonnes qui décide de l'ordre des points. Si k'_p est avant k'_q alors p est avant q dans l'ordre $l'k'r$ comme illustré dans la figure 5.13.

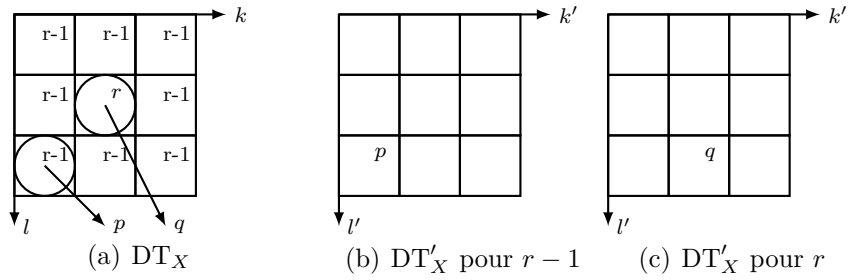


FIGURE 5.13 – Cas de l'équation 5.11b.

Cet ordre est le même que dans l'algorithme de référence à une configuration près, présentée dans la figure 5.14.

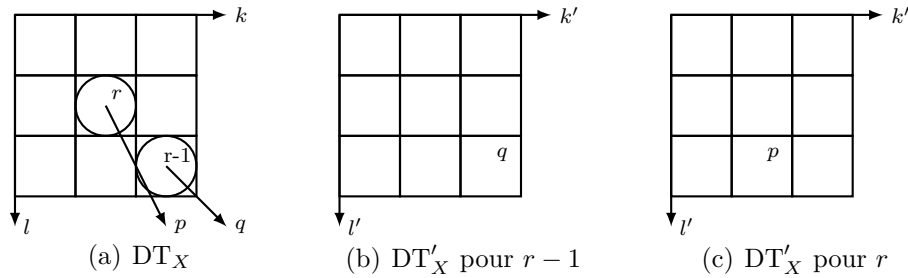


FIGURE 5.14 – Équation 5.11b.

Dans cette configuration, le point p est calculé avant le point q (car $l'_p = l'_q$ et $k'_p < k'_q$) alors que ce n'est pas le cas dans l'algorithme de référence puisque le point p est calculé avant le point q quand $DT_X(p) < DT_X(q)$ (voir équation 5.3a). Ce cas particulier est un problème à résoudre dans l'évaluation de la simplicité du point p . La fonction de simplicité est donc modifiée et l'on fait deux hypothèses sur l'état du point q qui est encore inconnu alors que dans *rlk* il est déjà calculé :

- le point q fait partie de l'axe médian (donc point non simple) alors il est nécessaire de contraindre le point p ,
- le point q ne fait pas partie de l'axe médian.

Algorithme 11: Algorithme de simplicité avec gestion du cas particulier.

Données : Image binaire X , r rayon de p , colonne k de p , ligne l de p , DT'_X

Résultat : 0 ou 1

```

1 si  $\bar{x}_4$  ET  $k > 0$  ET  $l + r \leq ls$  ET  $DT'_X[l + r][k - 1] > r$  alors
2   |  $x_4 \leftarrow 1$ ;
3 si  $\bar{x}_5$  ET  $k > 0$  ET  $l + r \leq ls$  ET  $DT'_X[l + r][k - 1] > r$  alors
4   |  $x_5 \leftarrow 1$ ;
5 si ( $\bar{x}_1$  ET ( $x_2$  OU  $x_3$ )
6   +  $\bar{x}_3$  ET ( $x_4$  OU  $x_5$ )
7   +  $\bar{x}_5$  ET ( $x_6$  OU  $x_7$ )
8   +  $\bar{x}_7$  ET ( $x_8$  OU  $x_1$ )) = 1 alors
9   | retourne 1;
```

La dernière partie de l'équation 5.11c ne propose pas de cas particulier. Si les lignes et les colonnes translatées sont égales, les points sont traités naturellement dans l'ordre croissant de leur valeur de distance. La figure 5.15 représente une de ces configurations.

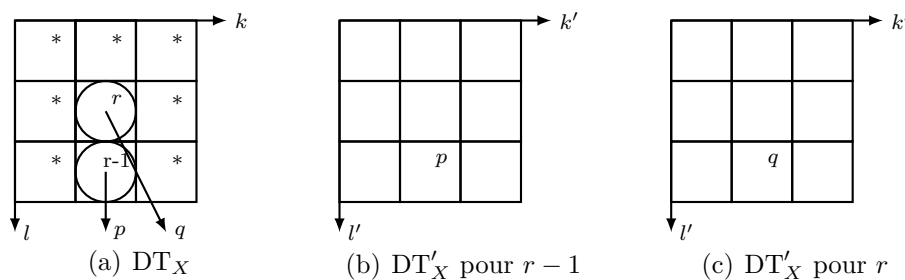


FIGURE 5.15 – Cas de l'équation 5.11c.

L'algorithme de squelettisation 12 met en évidence les trois boucles principales : lignes translatées, colonnes translatées et rayons de disques.

Les variables p et p' représentent respectivement le point et le point translaté. La ligne numéro 5 de l'algorithme affecte à la variable p les coordonnées du point dans la carte de distance DT_X grâce au point p' situé sur la carte de distance translatée DT'_X .

La condition de la ligne numéro 6 interdit le calcul du point si il est un centre de disque intermédiaire. Nous appelons disque intermédiaire, un disque complètement caché sous un autre. Attention, cette condition ne s'applique pas seulement aux points de l'axe médian. Par exemple un point p dont la valeur de la carte de distance est 1 peut être caché sous un disque dont le centre est un point de l'axe médian. Mais pour autant le point p ne cache pas forcément d'autre centre de disque. p est donc ce qu'on appelle un centre de disque non intermédiaire.

Pour savoir si le point n'est pas le centre d'un disque intermédiaire on inspecte le centre "présumé" du disque de rayon suivant. Si la valeur de distance du centre "présumé" du disque de rayon suivant est strictement inférieure à la valeur point courant plus 1, alors ce point courant n'est pas un centre de disque intermédiaire et on valide la condition.

La figure 5.16 illustre le cas de figure où le point n'est pas un centre de disque intermédiaire car la valeur de distance r du centre "présumé" du disque de rayon supérieur est de 0 soit strictement inférieur à r .

De cette manière, la gestion de la mémoire tampon se fait différemment d'une méthode de squelettisation classique. C'est-à-dire que lorsque l'on passe la condi-

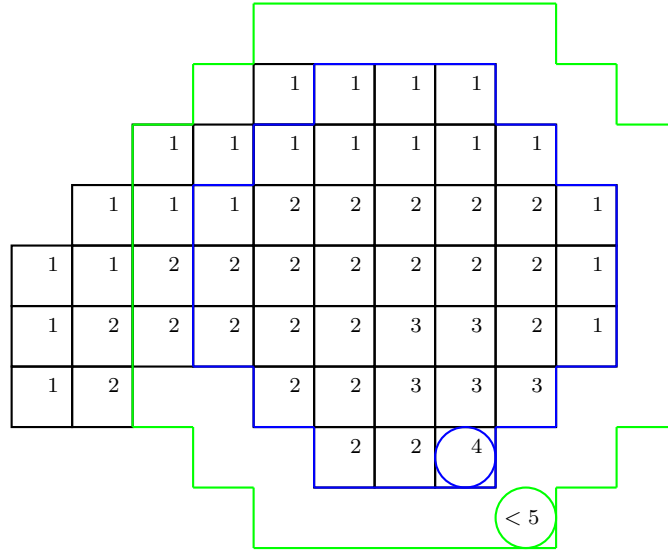


FIGURE 5.16 – Cas de figure où le point n n'est pas un centre de disque intermédiaire.

tion de la ligne 6, on va calculer dans un même temps tous les centres des disques intermédiaires.

Pour mieux nous rendre compte de l'ordre de calcul des points, regardons la figure 5.17. Dans cette figure, les valeurs indiquées correspondent au numéro de l'itération. La distance utilisée dans cet exemple est d_{oct} .

Cette figure nous montre que lors du balayage de l'image, certains pixels sont calculés plus tard. C'est le cas lors des itérations 23, 24, 31, 32, 36, 37, 38 et 39. Prenons l'exemple de cette dernière itération 39. Lors du balayage de ce point un dépilement des disques, dont le centre est situé en ce même point, est effectué. Pour chacun de ces centres de disques, par ordre croissant de rayon, une vérification de l'appartenance au squelette (appartenance à l'axe médian et simplicité du point) est réalisée.

Cette figure illustre également les différents vecteurs de dé-translation des centres des disques. Par exemple, au point visité à la 39^{ème} position se trouve le centre translaté de rayon 4 (selon d_{oct}). Lorsque ce point est identifié comme un point du squelette, il est enregistré à sa position dé-translatée (c'est à dire ici, au bout de la flèche).

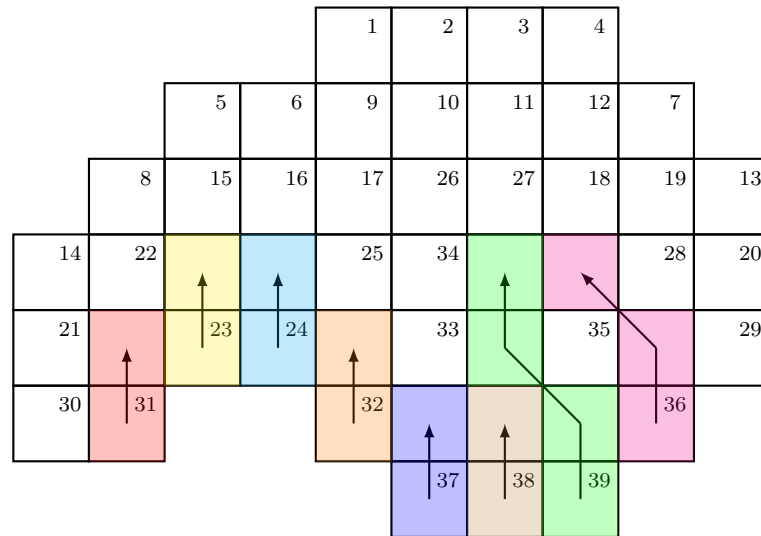


FIGURE 5.17 – Ordre de calcul de l'algorithme $l'k'r$.

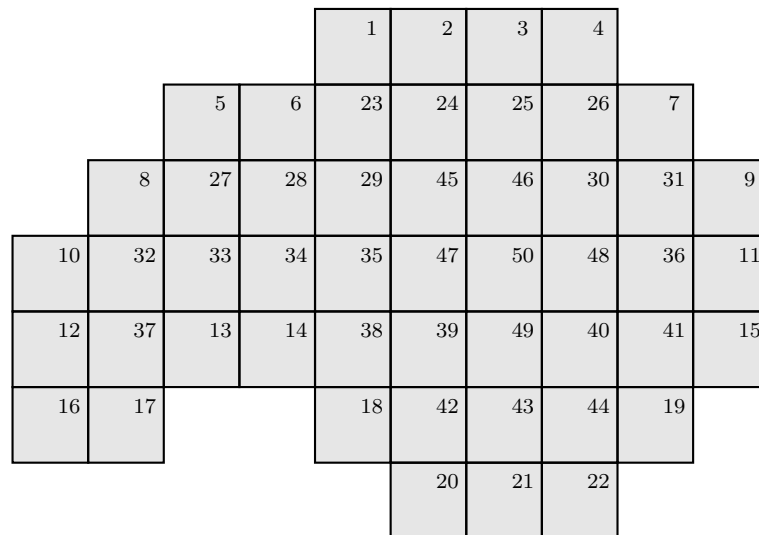


FIGURE 5.18 – Ordre de calcul de l'algorithme rlk .

Algorithme 12: Algorithme ligne'/colonne'/rayon ($l'k'r$)**Données :** Image binaire X , DT'_X , MA'_X **Résultat :** Squelette Sk

```

1 pour  $l' \leftarrow 0$  to  $K - 1$  faire
2   pour  $k' \leftarrow 0$  to  $L - 1$  faire
3      $p' \leftarrow (l', k')$ 
4     pour  $r \leq DT'_X(p')$  faire
5        $p \leftarrow p' - \overrightarrow{t(r)}$ 
6       si  $DT'_X(p + \overrightarrow{t(r+1)}) \leq r$  alors
7         si  $MA'_X(p') \neq r$  et  $p$  est simple selon l'algorithme 11 alors
8            $Sk_X(p) \leftarrow 0$ 

```

5.4 Résultats

Cette section présente les différents résultats de la squelettisation. Les trois algorithmes rlk , $l'rk'$ et $l'k'r$ produisent exactement les mêmes résultats. Afin de vérifier l'exactitude des résultats, nous procédons lors des tests à une comparaison entre les résultats obtenus avec les algorithmes rlk et $l'k'r$. Seuls les temps de calculs sont différents. C'est pourquoi la courbe des temps d'exécution de $l'k'r$ est présentée avec la courbe des temps d'exécution de l'algorithme rlk .

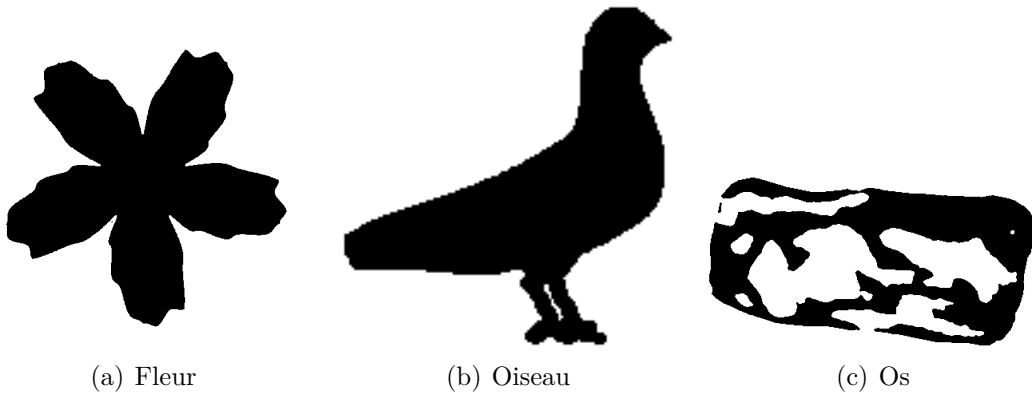


FIGURE 5.19 – Images originales binaires.

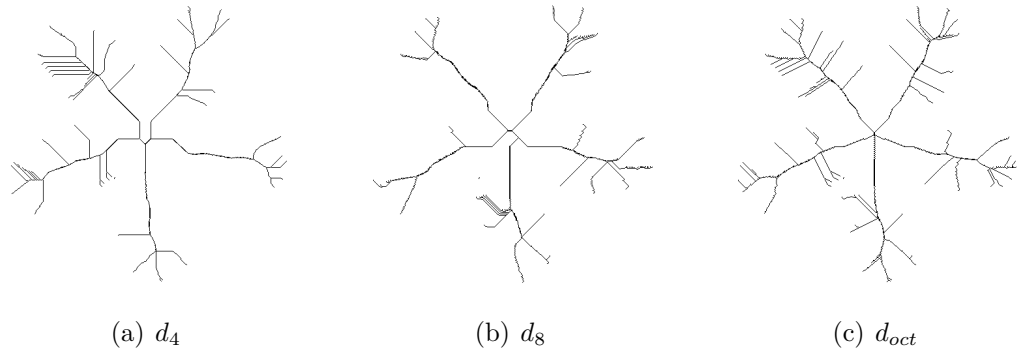


FIGURE 5.20 – Squelette de la fleur.

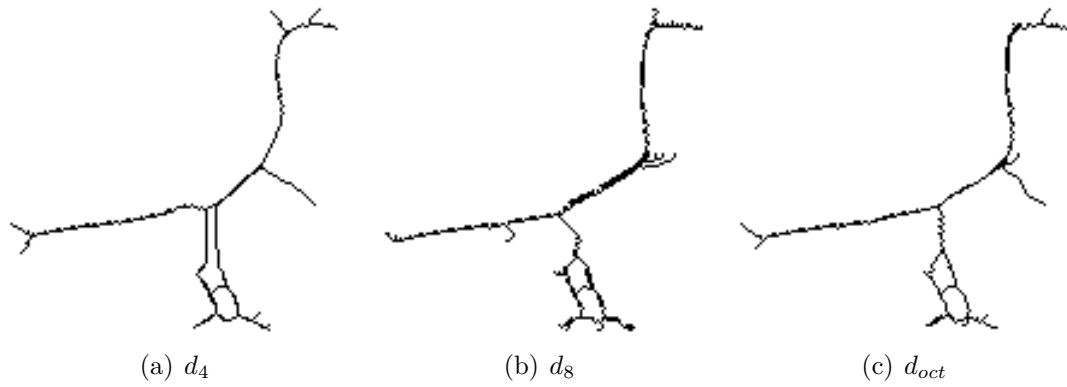


FIGURE 5.21 – Squelette de l'oiseau.

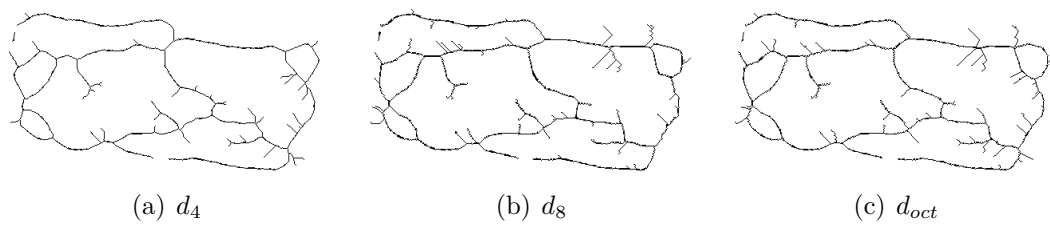


FIGURE 5.22 – Squelette de l'os.

Sur les squelettes résultants nous pouvons voir que :

- le squelette n'est pas fin partout dû à l'axe médian préservé dans le squelette,

- le squelette présente des sortes de petites ondulations dû à la reconnexion des points de l'axe médian.

En ce qui concerne l'épaisseur du squelette cela est normal puisque notre algorithme considère les points de l'axe médian comme point d'ancrage du squelette. Ce qui permet au squelette d'être centré dans la forme et de ce fait d'être entièrement restructurable grâce à la conservation des valeurs de la carte de distance dans le squelette.

Concernant les sortes de petites ondulations présentent sur le squelette, cela est dû à la manière de balayer l'image. Si notre algorithme de référence *rlk* calcule un squelette présentant des ondulations c'est parce qu'il n'y a pas de balayage rétrograde (qui supprimerait ces ondulations).

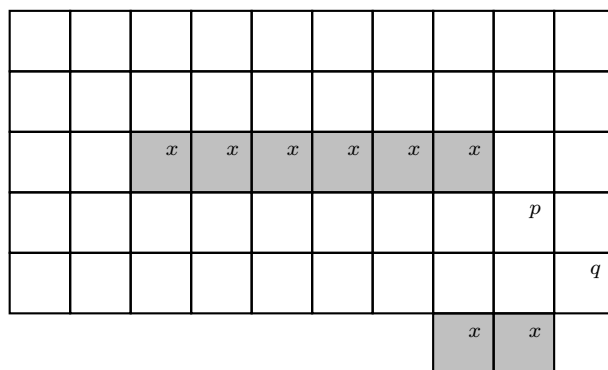


FIGURE 5.23 – Exemple de squelette présentant une ondulation.

Deux solutions ont été apportées pour éviter ces ondulations et rendre le squelette plus fin.

5.4.1 Augmentation de la résolution de l'image

En ajoutant des points entre les points existants nous augmentons la résolution (comme le fait Gilles Bertrand pour déterminer l'axe médian [19] ou Weixin Gong pour extraire le squelette [40]) de l'image et, de ce fait, multiplions les dimensions de l'image par 4. La figure 5.24 montre un exemple de points d'une image (points noir) et les points rajoutés (points blanc).

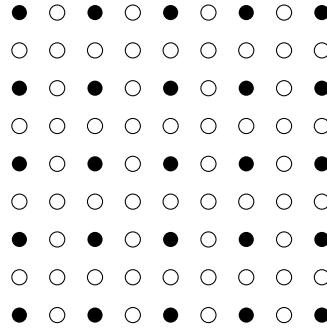


FIGURE 5.24 – Exemple de points en haute résolution.

Dans la figure 5.25 nous identifions les points de la maille. Les points a, c, g et i (en gris) ont les valeurs de l'image originale.

a	b	c
d	e	f
g	h	i

FIGURE 5.25 – Identification des points sur la maille dérivée.

Un point intermédiaire sur une ligne est ajouté lorsque que ses deux voisins (gauche et droite) appartiennent à la forme X . De la même manière, un point intermédiaire sur une colonne est ajouté que ses deux voisins (haut et bas) appartiennent à X . Enfin, un point est ajouté sur la ligne et la colonne intermédiaire (situation du pixel e) lorsque ses deux voisins diagonaux ou ses deux autres voisins diagonaux appartiennent à X :

$$\begin{aligned}
 b &= a \text{ ET } c \\
 d &= a \text{ ET } g \\
 e &= (a \text{ ET } i) \text{ OU } (c \text{ ET } g)
 \end{aligned}
 \tag{5.13}$$

De ce fait, une bande de pixels horizontale ou verticale devient une bande de

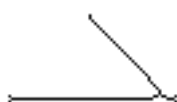
pixels impaire. L'amincissement d'une telle bande de pixels permet d'obtenir un squelette fin.

L'inconvénient de cette méthode est le temps de calcul. Car si on multiplie la taille de l'image par 4, on augmente également le temps de calcul. De plus, il existe encore des cas où le squelette n'est pas fin. La figure 5.26 représente une forme constituée de 2 barres, l'une est horizontale et l'autre a une direction diagonale.

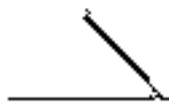


FIGURE 5.26 – Forme quelconque

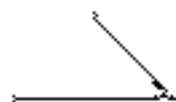
La figure 5.27 présente les squelettes, calculés avec les distances d_4 , d_8 et d_{oct} , de l'image 5.26. On constate que les squelettes calculés à partir de la carte de distance d_8 et d_{oct} ne sont pas fins sur la partie diagonale de la forme. Sur cet exemple le phénomène est moins visible sur le squelette obtenu avec d_{oct} .



(a) Squelette d_4



(b) Squelette d_8

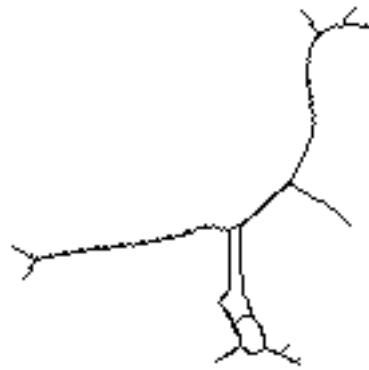


(c) Squelette d_{oct}

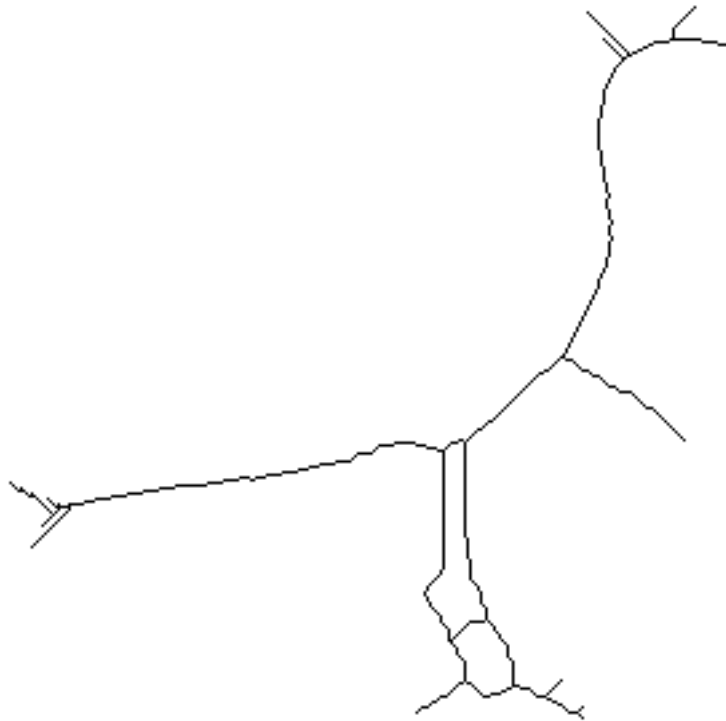
FIGURE 5.27 – Squelettes de la forme quelconque présentée par la figure 5.26.

Ceci est dû à la position des nouveaux points insérés dans la grille.

La figure 5.28 présente 2 squelettes de l'oiseau (présenté sur la figure 5.19) calculés à partir de la carte de distance d_4 . Le premier squelette est calculé dans sa résolution originale tandis que le second résultat est calculé après avoir augmenté la résolution. Dans ces résultats présentés, la proportion des images a été conservée.



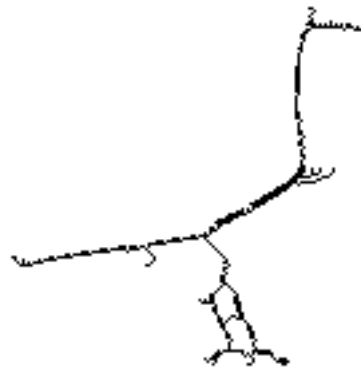
(a) résolution normale



(b) résolution augmentée

FIGURE 5.28 – Squelettes d_4 de l'oiseau

La figure 5.29 présente 2 squelettes de l'oiseau (présenté sur la figure 5.19) calculés à partir de la carte de distance d_8 .



(a) résolution normale



(b) résolution augmentée

FIGURE 5.29 – Squelettes d_8 de l'oiseau

La figure 5.30 présente 2 squelettes de l'oiseau (présenté sur la figure 5.19) calculés à partir de la carte de distance d_{oct} .



(a) résolution normale



(b) résolution augmentée

FIGURE 5.30 – Squelettes d_{oct} de l'oiseau

Les squelettes obtenus après un agrandissement sont visiblement plus fins que lorsque l'agrandissement n'a pas eu lieu. Ceci est d'autant plus vrai pour les squelettes calculés avec d_4 . En ce qui concerne les squelettes calculés à partir de d_8 et d_{oct} il est nécessaire de pratiquer un dernier amincissement pour obtenir un

squelette complètement fin.

5.4.2 Dernier amincissement

Une autre manière d'obtenir un squelette plus fin est d'appliquer un dernier balayage d'affinement du squelette. Cette méthode supprime les “doubles points” de l'axe médian et de ce fait le squelette n'est plus totalement bien centré dans la forme et plus entièrement reconstituable. De plus, ce dernier balayage risque de supprimer des morceaux entiers du squelette si la seule condition de suppression des points est la simplicité du point.

Afin de ne pas supprimer des morceaux de squelette nous allons détecter les points terminaux de celui-ci. Un point terminal est un point qui ne possède qu'un seul voisin (définition 13). Ces points terminaux seront utilisés comme point d'ancrage du squelette et non plus les points de l'axe médian comme précédemment.

La figure 5.31 illustre les squelettes avant et après le balayage d'amincissement.

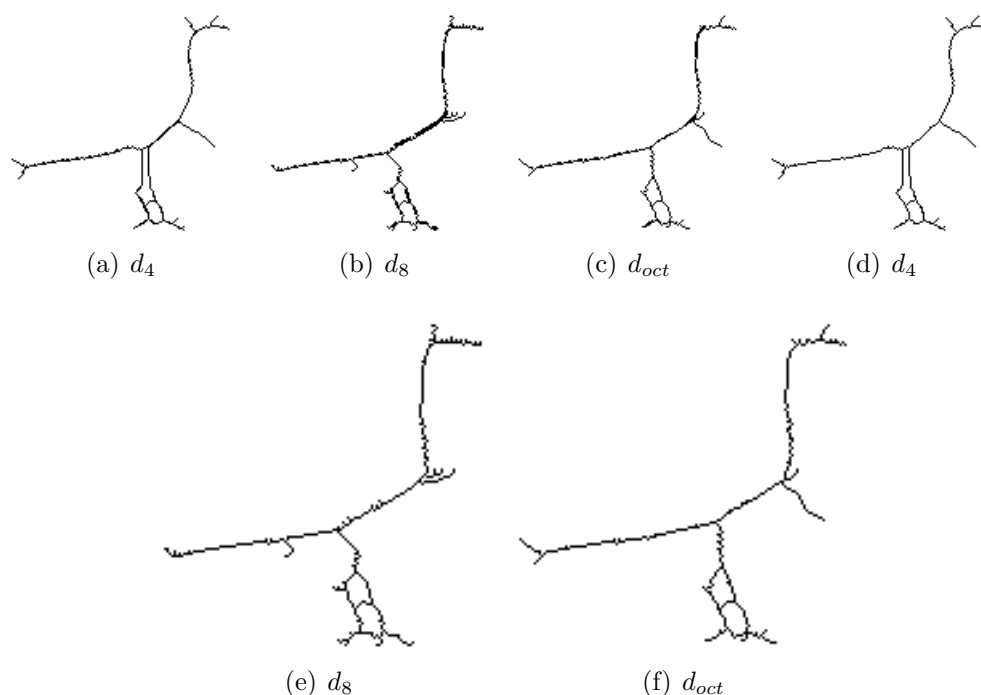


FIGURE 5.31 – Squelettes de l'oiseau (a, b et c) et squelettes amincis (d, e et f).

On peut alors combiner le fait d'augmenter la résolution et d'amincir une der-

nière fois la forme comme présenté dans la figure 5.32.

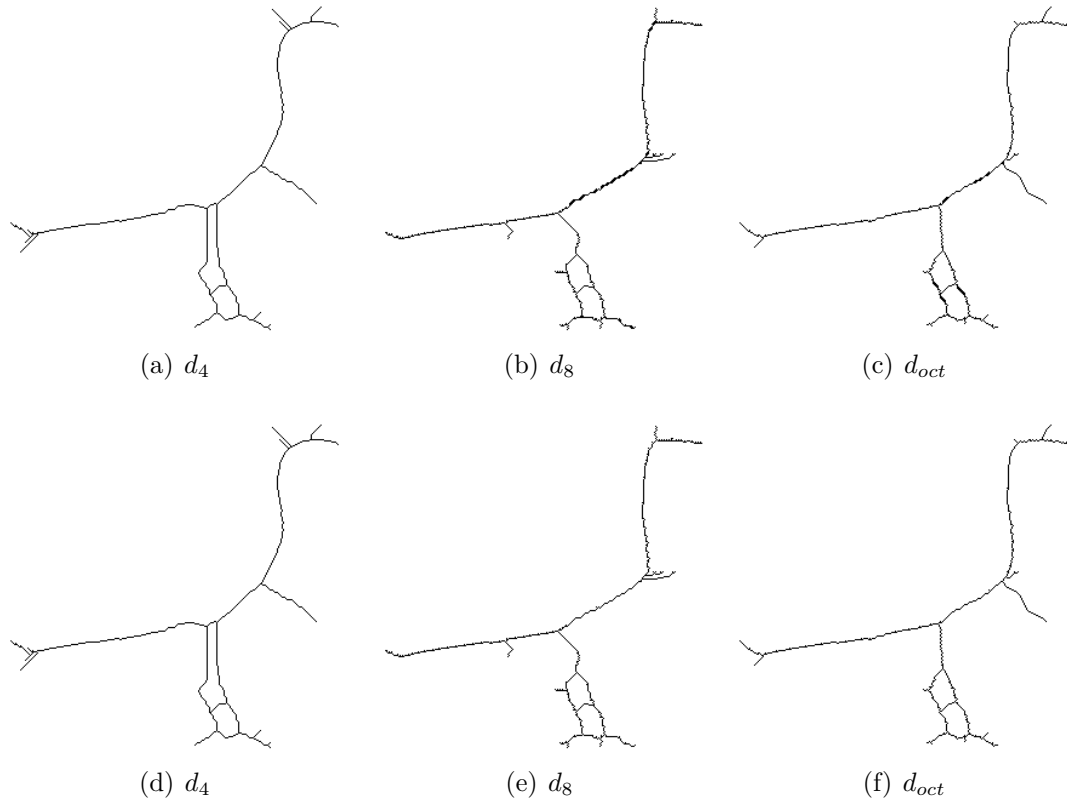


FIGURE 5.32 – Squelettes de l’oiseau (a,b et c) et squelettes amincis (d,e et f) avec une résolution augmentée.

5.4.3 Temps de calcul

Cette sous-section présente les temps d'exécution des différents algorithmes. Ces 3 graphiques présentent les temps d'exécution des algorithmes de squelettisation de référence rlk et en un balayage de l'image $l'k'r$.

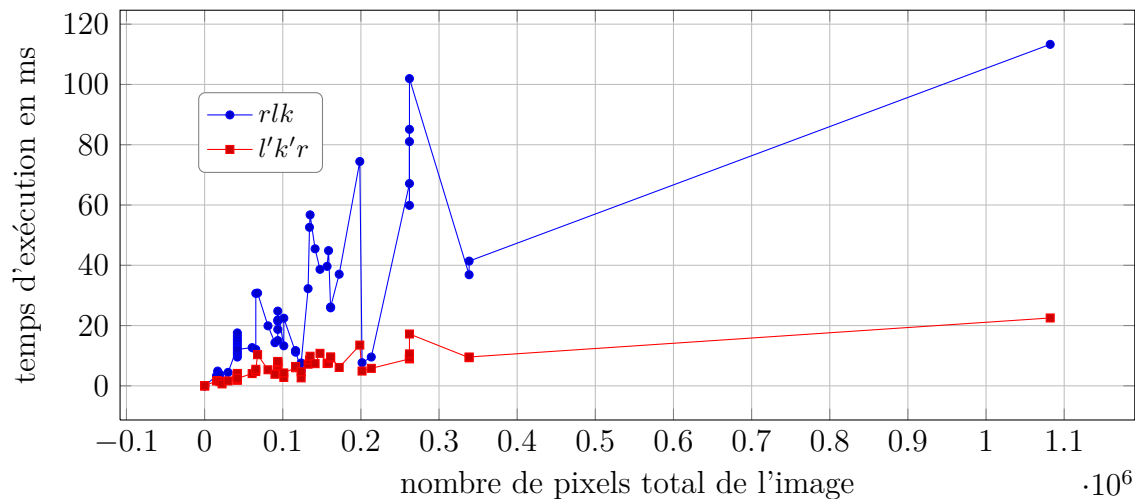


FIGURE 5.33 – Temps de calcul des algorithmes de squelettisation en d_4 .

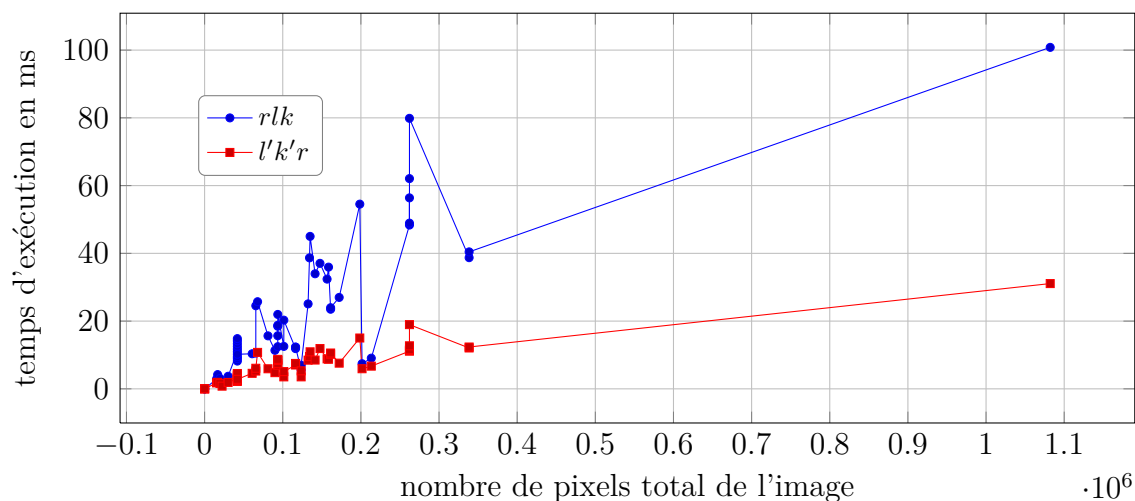
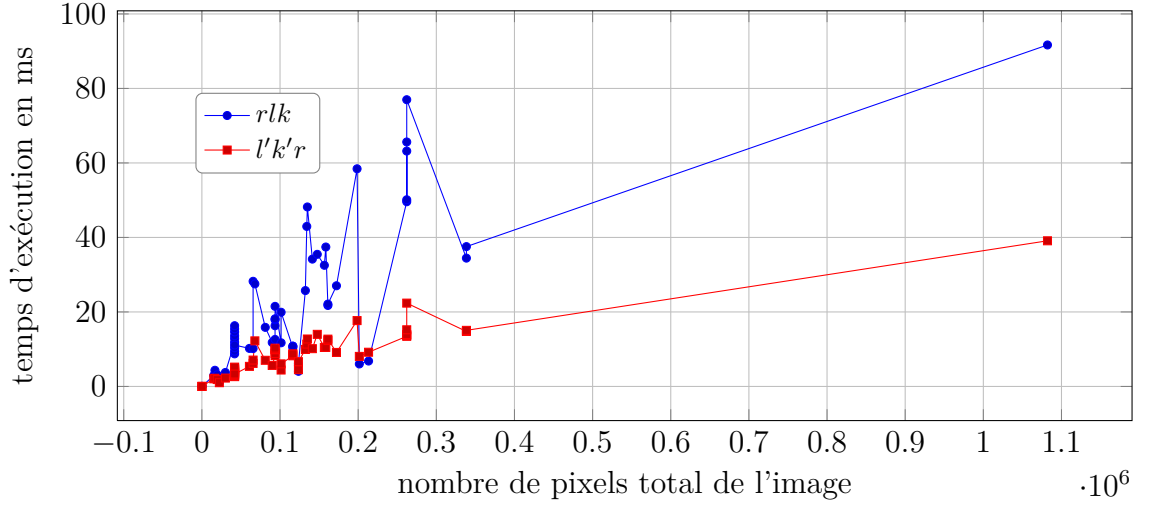


FIGURE 5.34 – Temps de calcul des algorithmes de squelettisation en d_8 .


 FIGURE 5.35 – Temps de calcul des algorithmes de squelettisation en d_{oct} .

On remarque que plus il y a de pixels dans l'image, plus l'écart entre les temps d'exécution des algorithmes rlk et $l'k'r$ est important. À l'inverse, quand il y a peu de pixels à traiter, la différence est minime.

Ceci s'explique par le fait que l'algorithme $l'k'r$ optimise les traitements effectués sur les points. C'est-à-dire que certains points sont "ignorés" à un moment pour être ensuite traités dans leur voisinage translaté (comme illustré dans la figure 5.17). Ce qui revient à optimiser l'accès mémoire de chaque point.

De plus, l'algorithme rlk parcourt r fois l'image alors que l'algorithme $l'k'r$ ne la parcourt qu'une seule fois. Donc plus l'image contient de points, plus notre algorithme $l'k'r$ est plus rapide que l'algorithme de référence rlk .

La complexité spatiale de notre algorithme $l'k'r$ est la suivante :

$$\begin{aligned}
 S_{DT'_X}(k) &= 2k + 2 \\
 S_{MA'_X}(k) &= 2k + 2 \\
 &= (2k + 2) \lceil \log_2(r_{max} + 1) \rceil \\
 S_{SK'_X}(k) &= r_{max} k
 \end{aligned} \tag{5.14}$$

Pour calculer la carte de distance translatée et l'axe médian, seules deux lignes (plus deux pixels dans le cas d'utilisation de \mathcal{N}_2) sont nécessaires. La figure 5.36 illustre, dans les deux types de voisinages, quelles sont les lignes (en rouge) stockées en mémoire afin de calculer le point p . Dans le cas d'utilisation du voisinage de

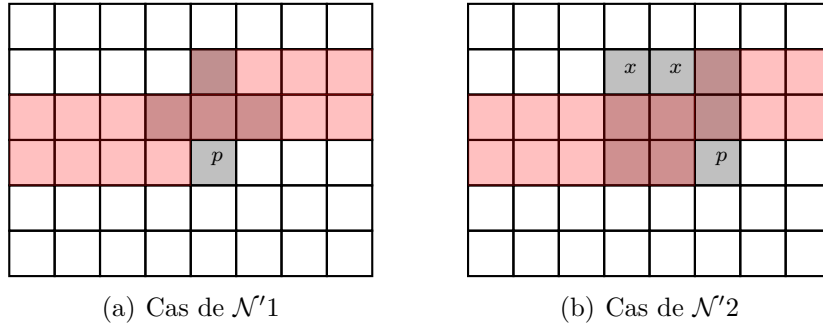


FIGURE 5.36 – Information stockée en mémoire pour calculer la distance du point p .

type 2, deux pixels supplémentaires (notés x) sont nécessaires de stocker afin de connaître toutes les valeurs du voisinage du point p .

La complexité algorithmique des algorithmes rlk et $l'k'r$ s'exprime de la façon suivante :

$$\begin{aligned} rlk &= k \times l \times r_{max} \\ l'k'r &= k \times l + \sum r \end{aligned} \tag{5.15}$$

5.4.4 Discussion

Dans la suite de ce travail nous allons utiliser notre algorithme de squelettisation pour extraire des caractéristiques osseuses. Nous n'allons donc pas exploiter toutes les propriétés du squelette :

1. topologiquement équivalent à la forme,
2. centré dans la forme,
3. fin,
4. restructurable.

On a vu précédemment que la propriété 2 n'est pas, dans tous les cas, compatible avec la propriété 3. En effet lorsque la forme a une épaisseur paire, le squelette est soit bien centré dans la forme mais a une épaisseur de 2 pixels soit a une épaisseur d'un pixel mais pas rigoureusement centré dans la forme.

Dans ce même contexte, les propriétés 3 et 4 ne sont pas compatibles car l'aminçissement d'un squelette se traduit par la suppression des points de l'axe médian dans le squelette. Si lors de la reconstruction de la forme, il manque des points de l'axe médian alors on perd des points situés sur le bord de la forme.

Les squelettes calculés grâce à notre algorithme sont restructurables. C'est-à-dire qu'à partir du squelette, il est possible de reconstruire la forme initiale puisque nous avons sauvegardé l'information de distance. Nous ne présenterons pas ici les résultats de la reconstruction puisque la forme reconstruite est exactement identique à la forme originale.

Il est cependant notable que la forme reconstruite à partir du squelette affiné (après un aminçissement ultime) présente quelques différences. La figure 5.37 présente la forme originale (jaune) avec son squelette (noir et rouge) et son squelette affiné (noir).



FIGURE 5.37 – Forme avec son squelette et son squelette affiné.

La figure 5.38 présente les différences entre la forme reconstruite à partir de squelette (calculé à partir de d_8) affiné (gris) et la forme originale (gris et noir). Cette figure montre donc en noir les pixels perdus lors de la reconstruction à partir du squelette affiné.



FIGURE 5.38 – Forme reconstruite à partir du squelette affiné.

On constate sur cet exemple que l'on perd peu d'information. L'invariance en rotation est une propriété, que l'on peut ajouter au squelette, qui est liée avec le fait que le squelette est centré dans la forme. Nous avons testé cette

invariance avec une image binaire d'os trabéculaire (figure 5.39) à laquelle on a effectué 3 rotations.

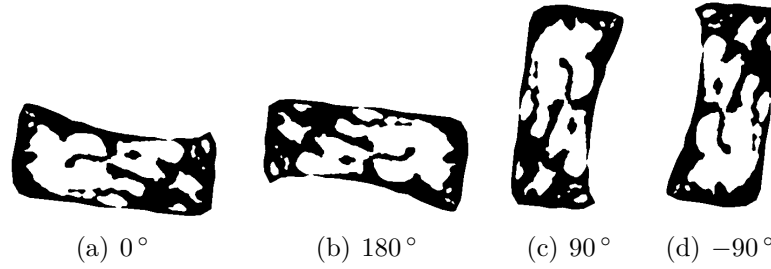
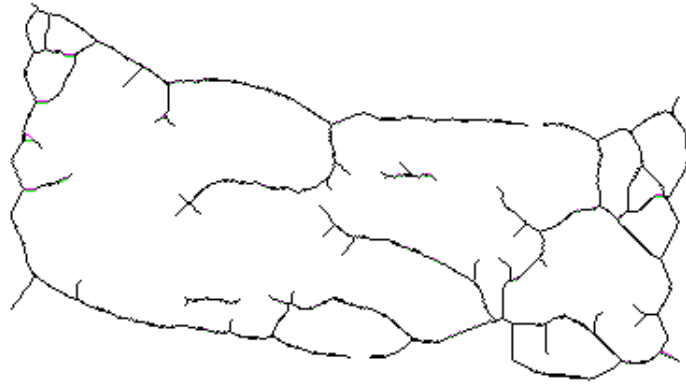


FIGURE 5.39 – Images originales binaires.

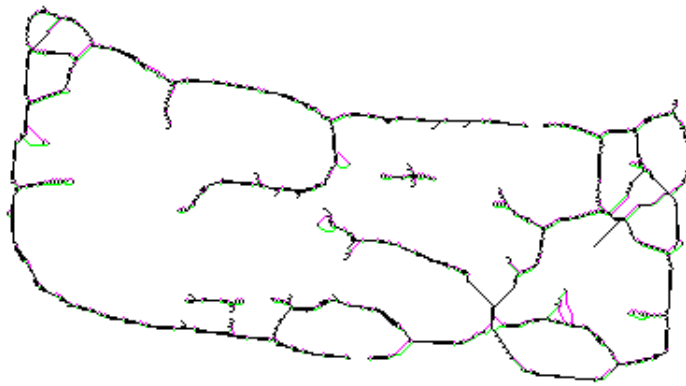
Les squelettes de ces 4 images sont ensuite calculés puis on a réappliqué une rotation de telle sorte que les squelettes se trouvent dans la position 0°. Le but de cette manipulation est de visionner les différents résultats obtenus sur la même image sous différentes orientations.

La figure 5.40 présente la différence des squelettes calculés, selon les distances d_4 , d_8 et d_{oct} , à partir de l'image non tournée et des squelettes calculés à partir des images orientées à 90° puis ré-orientées à 0°.

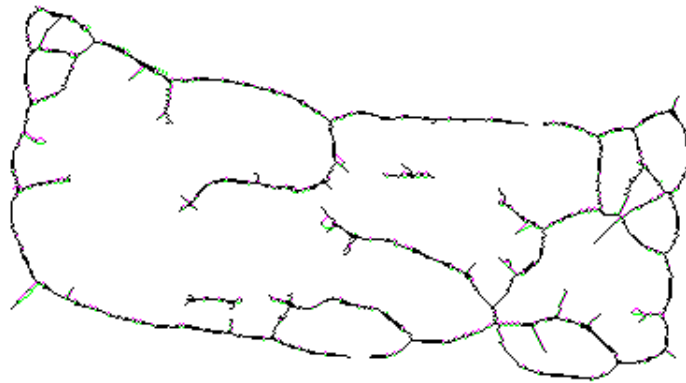
La version imprimée de ce document ne permet pas de bien visualiser les différences. Pour mieux les apprécier, se référer à la version électronique. Les pixels noirs correspondent à un résultat identique alors que les pixels roses correspondent au squelette de l'image sans rotation et les pixels verts correspondent au squelette de l'image qui a subi la rotation.



(a) d_4



(b) d_8



(c) d_{oct}

FIGURE 5.40 – Différences de squelettes après rotation de 90° .

Sur cette figure, pour les trois distances utilisées il y a des différences locales. On note que ces différences apparaissent le plus sur les barbules du squelette.

Afin de mieux se rendre compte des différences, la figure 5.41 présente un morceau du squelette. Pour chaque rotation, une comparaison est présentée :

- rotation à 90° : (a), (b), et (c),
- rotation à -90° : (d), (e) et (f),
- rotation à 180° : (g), (h) et (i).

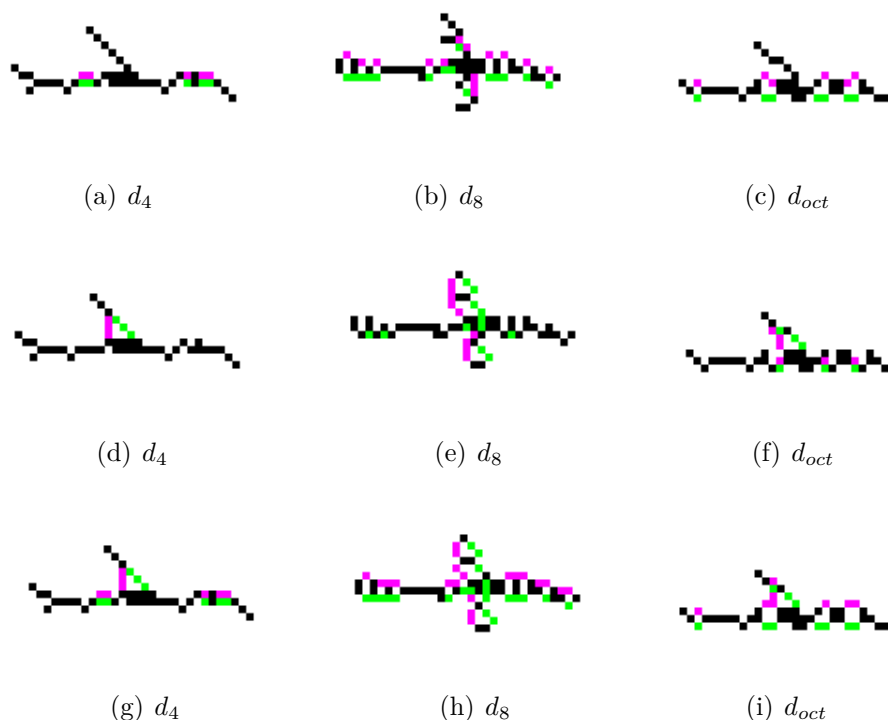


FIGURE 5.41 – Différences entre squelettes avant et après rotation.

Les points de l'axe médian sont préservés dans le squelette. Ces points sont donc invariants en rotation puisque qu'ils sont les centres des boules maximales incluses dans la forme. Cependant les autres points reliant ces points de l'axe médian sont dépendants de la manière dont est balayée l'image. C'est à dire l'ordre dans lequel on supprime ou pas un point et donc leur ordre d'apparition dans le balayage. Ce phénomène est le plus visible dans l'exemple de la figure 5.41 pour le cas de la rotation à 90° (première ligne). Sur cet exemple, les pixels roses ont été visités

avant les pixels verts dans l'image non orientée alors que les pixels verts sont visités avant les roses quand l'image est orientée à 90° .

5.5 Conclusion

Dans ce chapitre a été présenté un algorithme de squelettisation nécessitant un seul balayage de l'image pour extraire le squelette.

Rappelons les principales propriétés du squelette :

1. homotope,
2. restructurable,
3. centré,
4. finesse.

Ces quatre propriétés ne peuvent pas toutes être obtenues par un algorithme de squelettisation. Par exemple, une forme d'épaisseur paire a soit un squelette centré et une épaisseur de deux pixels, soit le squelette est fin mais légèrement décentré. De même que si l'on souhaite obtenir un squelette fin, il ne sera pas entièrement restructurable (perte d'information sur les bords de la forme).

Cette raison nous pousse donc à faire un choix dans les propriétés que l'on souhaite conserver dans le squelette. L'ordre de priorité des propriétés que nous avons choisi est 1,2,3 puis 4.

Pour cela nous avons choisi un algorithme de référence qui est une méthode hybride : un amincissement contrôlé par la distance.

Cette méthode permet d'obtenir des squelettes topologiquement équivalents à la forme. Ils ne sont pas forcément fins puisque l'axe médian est conservé dans le squelette mais lorsqu'un dernier amincissement n'est pas effectué, le squelette est centré et restructurable. Si l'étape du dernier amincissement est appliqué, le squelette est fin mais pas complètement centré et restructurable (très peu de pixels sont perdus sur les bords de la forme).

La démarche adoptée a été de modifier pas à pas un algorithme de référence afin d'obtenir un algorithme nécessitant un seul balayage de l'image pour produire le squelette.

Pour ce faire, une modélisation de la dépendance entre pixels, a été proposée (sous forme d'ordre partiel). Nous avons ensuite proposé trois mises en œuvre séquentielles de cet ordre partiel. C'est le respect de ce dernier dans les trois

algorithmes qui garantit l'identité des résultats.

Alors que notre algorithme de référence (nommé *rlk*) parcourt toutes les valeurs de la carte de distance DT_X puis les lignes et les colonnes de l'image, l'algorithme final (nommé *l'k'r*) parcourt les lignes, les colonnes puis les valeurs de la carte de distance translatée DT'_X .

Ce processus de squelettisation en un balayage de l'image est plus rapide que l'algorithme de référence pour deux raisons. D'une part, les accès à la mémoire sont mieux localisés que ceux d'un autre algorithme. D'autre part chaque point est visité au maximum 2 fois.

L'algorithme en un balayage de l'image génère des résultats identiques, au pixel près, aux résultats de l'algorithme de référence. Ceci parce que l'on s'est attaché à respecter l'ordre partiel. Dans seulement 2 cas bien identifiés cet ordre n'est pas respecté. Ces cas particuliers sont traités séparément de manière à obtenir une cohérence de la dépendance entre pixels.

Conclusion

Dans cette partie a été présentée une méthode de squelettisation ne nécessitant qu'un seul balayage de l'image pour extraire le squelette.

L'information arrive des bords de la forme et se propage vers l'intérieur de celle-ci. C'est pour cette raison que les algorithmes de la littérature requièrent plusieurs balayages de l'image. Dans le cas contraire, l'information serait décentrée et faussée.

Notre but dans ce travail a été de produire des algorithmes propageant des informations du haut de l'image vers le bas de celle-ci.

Notre algorithme se base sur une méthode hybride : amincissement contrôlé par la distance. Cette méthode permet d'obtenir un squelette homotopique (suppression des points régie par une fonction de simplicité), restructurable et centré (l'axe médian étant conservé).

La première phase de ce travail consiste à calculer une carte de distance en un balayage de l'image. Pour cela, une translation des centres des disques a été effectuée. De cette manière, on obtient une carte de distance asymétrique DT'_X . Afin de trouver une carte de distance symétrique, un algorithme de dé-translation de la carte de distance a été proposé. Dans ce travail, les distances d_4 , d_8 et d_{ns} ont été développées.

La deuxième phase de ce travail a été d'identifier les points de l'axe médian sur la carte asymétrique. Comme des distances simples ont été utilisées, la recherche de l'axe médian est un ensemble de calculs locaux. Il n'y a donc pas de propagation d'information dans l'image.

La troisième et dernière phase de ce travail de squelettisation consiste à supprimer les points de la forme pour obtenir le squelette. Dans l'algorithme de référence *rlk*, l'information se propage de tous les bords de la forme vers l'intérieur de celle-ci. L'objectif étant de propager l'information du haut vers le bas de l'image, nous avons étudié la dépendance entre pixels (sous forme d'ordre partiel) afin d'obtenir

les mêmes résultats que ceux produits par l'algorithme de référence. Cet algorithme de référence balaye l'image selon les valeurs des rayons, les lignes puis les colonnes de l'image. Notre algorithme $l'k'r$ (en un balayage) parcourt les lignes et les colonnes de la carte de distance DT'_X puis les rayons tout en respectant l'ordre partiel. Ce respect de l'ordre partiel nous assure l'identité des résultats.

Ce traitement calcule le squelette avec un seul balayage de l'image. Il permet de calculer un squelette sur une image potentiellement non bornée temporellement (image en cours d'acquisition) ce qui n'est pas possible avec un algorithme de la littérature.

Les squelettes obtenus par notre algorithme $l'k'r$ sont identiques au pixel près aux squelettes calculés avec l'algorithme de référence rlk . L'empreinte mémoire est optimisée car ses accès sont mieux localisés que ceux de l'algorithme de référence.

L'animation qui suit (visible uniquement sur une version électronique avec Adobe Acrobat Reader) montre la progression du calcul du squelette selon l'algorithme de référence puis notre algorithme $l'k'r$.

La figure 5.42 présente les résultats à différentes itérations de l'algorithme de référence rlk .

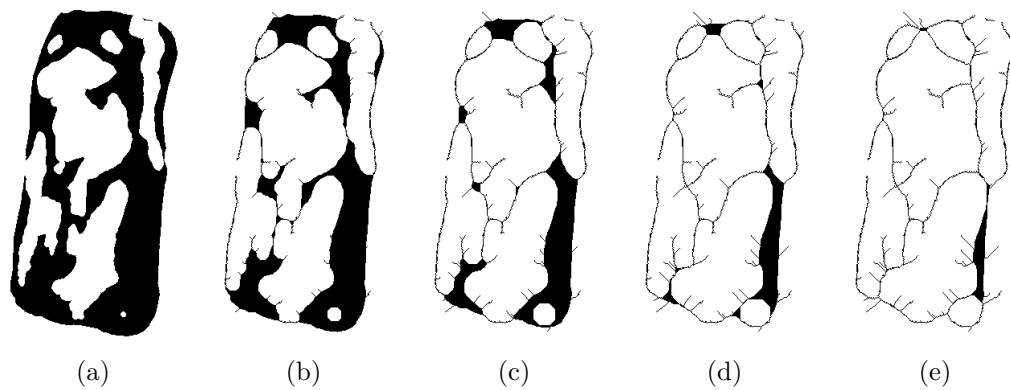


FIGURE 5.42 – Processus de squelettisation à différentes itérations selon l’algorithme de référence rlk .

La figure 5.43 présente les résultats à différentes itérations de l’algorithme en un balayage unique de l’image $l'k'r$.

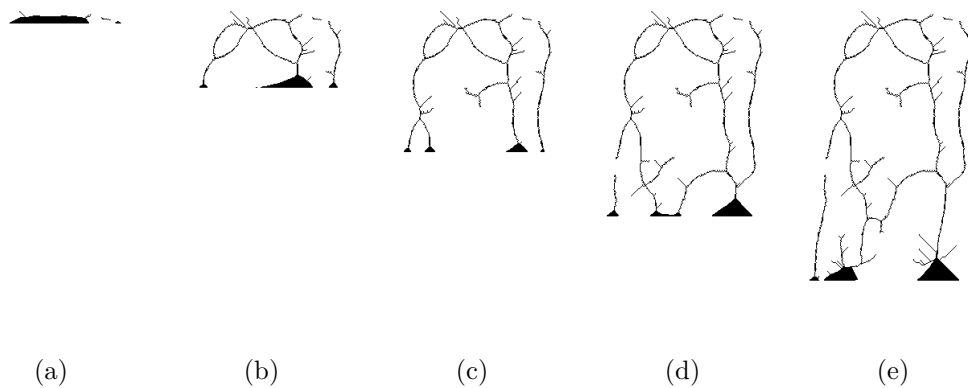


FIGURE 5.43 – Processus de squelettisation à différentes itérations notre algorithme en un balayage $l'k'r$.

Le calcul du squelette va nous permettre, dans la suite de ce travail, de caractériser l’architecture osseuse.

Troisième partie

Application biomédicale

Introduction

L'équipe *IVC* (Image et VidéoCommunication) à laquelle j'appartiens est engagée dans le projet régional de recherche Bioregos2.

Le corps Humain compte 206 os selon [73]. Entouré de muscles, ligaments, cartilages... l'os charpente notre corps. L'os a une double fonction :

- une fonction mécanique de résistance à la pesanteur, aux forces exercées par les muscles, etc.
- une fonction métabolique. L'os est en effet le premier réservoir en ions minéraux de l'organisme (calcium, phosphore notamment). La calcémie par exemple est uniquement régulée par départ ou arrivée d'ions calcium du tissu osseux vers le sang.

L'os assure également la protection de certains organes. Les os du crâne protègent le cerveau et la cage thoracique protège le coeur, les poumons, etc. De plus il permet la fabrication de cellules sanguines grâce à la moelle osseuse.

Il existe des maladies qui touchent les os, les principales sont :

- le cancer, qui est rarement primaire mais dont les métastases sont souvent osseuses,
- l'ostéogénèse imparfaite, autrement dit la “maladie des os de verres” qui est une fragilité excessive des os due à un défaut congénital,
- l'arthrose est une maladie qui touche les articulations, le cartilage et l'os intérieur. La surface du cartilage se fissure puis s'effrite, disparaît puis des excroissances osseuses se forment et gênent la mobilité.
- l'ostéoporose est une fragilité excessive du squelette due à une diminution de la masse osseuse et à l'altération de la microarchitecture osseuse.

L'ostéogénèse imparfaite est une maladie rare des os contrairement à l'arthrose et l'ostéoporose. Il est important d'étudier, par imagerie, la structure de l'os afin de connaître sa modification due à la maladie.

Dans ce travail nous recherchons à déterminer la qualité de l'os à partir de traitements de morphologie mathématique. La caractérisation de l'os a plusieurs

intérêts. Elle permet de connaître et de mieux comprendre la structuration trabéculaire et aussi d'apprécier la qualité osseuse afin, par exemple, de diagnostiquer des maladies comme l'ostéoporose. Nous travaillons avec le Professeur Yves Amourig, du Laboratoire d'Ingénierie Ostéo-Articulaire et Dentaire "LIOAD" INSERM UMR S 791, sur des os de la mâchoire. Dans cette partie est décrite une méthode d'extraction d'informations de la structure trabéculaire basée sur la squelettisation expliquée dans la partie précédente.

Le schéma 5.1 récapitule les différentes étapes du processus.

FIGURE 5.1 – Schéma du processus général.



L'acquisition peut se faire de plusieurs manières :

- en 2D par radiographie à rayons X ou par ultra sons,
- en 3D par microscanner (analyse *in vitro*) ou par "cone beam" (analyse *in vivo*).

L'analyse d'image se fait par apprentissage si l'on se trouve dans le cas *in vivo*. C'est à dire que l'on va imager le patient et pratiquer une classification (comme la densitométrie qui permet de calculer la Densité Minérale Osseuse *DMO*) à partir de ces radiographies. L'acquisition par "cone beam" de l'image est également utilisée [46] pour déterminer la qualité osseuse mais elle ne donne pas la même résolution qu'un microscanner.

Dans le cas *in vitro*, où l'on a biopsié l'os, on pourra appliquer des traitements d'images beaucoup plus précis et représentatifs de la structure osseuse.

Les détails de la phase d'acquisition sont expliqués dans le chapitre 6 ainsi que les pré-traitements. La phase de squelettisation a été présentée dans la partie précédente II. Enfin l'extraction de ces caractéristiques de l'architecture osseuse et la définition des ces caractéristiques sont présentées dans le chapitre 7.

Étude de l'os trabéculaire

Dans ce chapitre nous nous intéressons à l'étude de l'os trabéculaire. Dans un premier temps, nous identifions cet os (section 6.1). À cette occasion, nous identifierons sur notre volume d'images (fourni par le LIOAD) la zone trabéculaire.

Différentes caractéristiques sont extraites de l'os (par les chercheurs du LIOAD et autres) pour caractériser la micro-architecture osseuse. La section 6.2 présente différentes caractéristiques osseuses extraites de la littérature.

Dans une dernière section 6.3 sont présentés les étapes d'acquisition et les différents post-traitements effectués avant de pratiquer la squelettisation.

6.1 Qu'est-ce que l'os trabéculaire ?

L'os se compose en deux parties :

- Compacte que l'on appelle la zone corticale,
- Spongieuse que l'on appelle la zone trabéculaire.

L'os cortical, ou compact, se trouve sur la partie externe de l'os. Très dur et résistant il protège l'os trabéculaire se trouvant à l'intérieur et beaucoup plus fragile. L'os trabéculaire possède des espaces conjonctivo-vasculaires non minéralisés très vastes, ce qui caractérise un os spongieux. Il se compose par ailleurs d'un réseau particulièrement complexe de travées interconnectées. Cette zone trabéculaire est intéressante pour analyser les caractéristiques de l'os (sain ou malade). Par ailleurs, l'os cortical étant à peu près constant, son influence lors d'une projection radiographique peut être négligée [2].

Les figures 6.1 représentent une radiographie d'un segment de la mandibule d'un chien comprenant une dent et de l'os où l'on va analyser la zone trabéculaire. Cette zone est identifiée par un cube gris et la seconde image représente le volume 3D de cette zone trabéculaire reconstruite par le microscanner. Afin de mieux

visualiser cette zone trabéculaire, l'os cortical a été retiré.

La figure 6.2 représente la radiographie de la mandibule de chien et les lignes

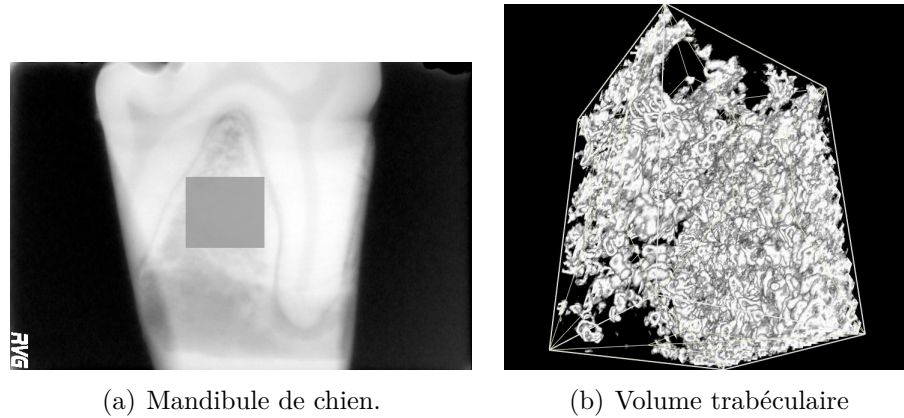


FIGURE 6.1 – Localisation de la zone d'intérêt et représentation 3D du réseau trabéculaire.

noires représentent 4 couches d'images 2D. Sur cette radiographie, les couleurs sont inversées et le noir représente les parties de plus forte densité, opaque aux rayons X (donc l'os et la dent).

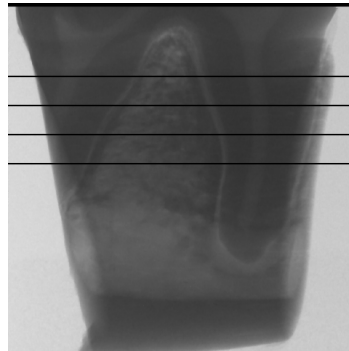


FIGURE 6.2 – Radiographie de la mandibule de chien.

Les images 6.3 représentent quelques images 2D (section) issues du volume de la mandibule de chien qui correspondent aux 4 traits noirs horizontaux de l'image 6.2.

Les images 6.4 représentent les mêmes images 6.3 correspondant à l'os trabéculaire de la région centrale des images précédentes. Le praticien a supprimé les

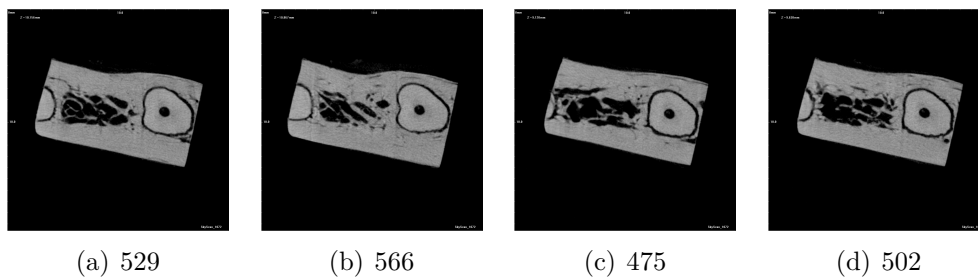


FIGURE 6.3 – Images μ -CT de l'os trabéculaire et de l'os cortical.

régions de l'os cortical.

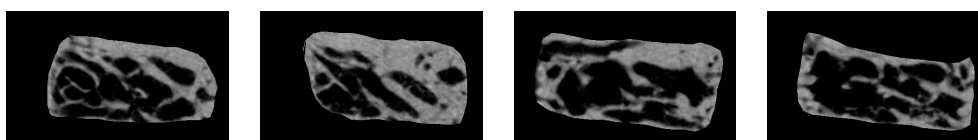


FIGURE 6.4 – Os trabéculaire.

Ces dernières régions d'intérêts ont été binarisées (avec l'aide du logiciel *ImageJ* avec un seuil choisi par le praticien) comme présentées en figure 6.5



FIGURE 6.5 – Images binarisées d'os trabéculaire (blanc = os).

6.2 Comment caractériser l'os ?

La micro-architecture osseuse est un bon indicateur de la qualité de l'os. Les informations de caractérisation de cette micro-architecture sont très utiles pour détecter la plupart des maladies de l'os comme l'ostéoporose, pour assurer le suivi des cicatrisations osseuses (avec ou sans l'utilisation des biomatériaux) ou de greffes osseuses et de comprendre les propriétés mécaniques de l'os.

L'étude de la micro-architecture osseuse est basée sur des mesures de largeur, longueur des trabécules ainsi que leur embranchement et terminaison pour déterminer/qualifier leur organisation spatiale.

Un grand nombre de travaux a été effectué pour extraire ces caractéristiques et qualifier la structure osseuse. Dans [68] sont répertoriés différents types d'acquisition d'images (radiographie, tomographie, microtomographie, IRM) et leur méthode d'extraction de caractéristiques.

Le tableau 6.1 représente les caractéristiques osseuses générales extraites et étudiées aujourd'hui avec leur abréviation (du nom anglais) et leur unité de mesure. Ces caractéristiques sont actuellement calculées grâce au logiciel *CTan* utilisé par les odontologues du laboratoire LIOAD au sein de l'université de Nantes.

Paramètre	Abréviation	Unité
Volume osseux/Volume Tissulaire	BV/TV	%
Nombre trabéculaire	Tb.N	$1/mm$
Épaisseur trabéculaire	Tb.Th	mm
Séparation trabéculaire	Tb.Sp	mm
Longueur totale du squelette	TSL	mm
Nombre de noeud/Volume trabéculaire	NN/TV	N/mm^2
Noeud/Terminaison	N/T	%
Volume de la moelle	MaSV	N/mm^3
Dimension fractale	D	-
Facteur de forme de l'os trabéculaire	Tb.Pf	-
Index d'interconnexion	ICI	-
Nombre d'Euler	E/TV	mm^2

TABLE 6.1 – Caractéristiques osseuses communément étudiées.

Les paramètres de Parfitt [66] sont les plus anciens et largement utilisés [14, 20, 43, 69]. Une nomenclature de ces paramètres osseux a été produite plus tard [65].

BV/TV représente le volume trabéculaire osseux. C'est-à-dire qu'après classification de l'os (cortical et trabéculaire), ce paramètre est calculé en divisant le volume total de l'os par le volume de l'os trabéculaire. Il est d'une importance

primordiale pour comprendre la salubrité de l'os.

Le nombre trabéculaire $Tb.N$ ne correspond pas exactement au nombre de trabécules mais une estimation du nombre de trabécules. On le calcule en divisant le volume trabéculaire osseux par l'épaisseur trabéculaire : $Tb.N = (BV/TV)/Tb.Th$ [65].

L'épaisseur trabéculaire $Tb.Th$ est calculée à partir de l'aire ($B.Ar$) et du périmètre ($B.Pm$) trabéculaire : $Tb.Th = 1.199 \times B.Ar/2/B.Pm$ [29].

L'espace inter-trabéculaire $Tb.Sp$ est calculé à partir du nombre trabéculaire et de l'épaisseur trabéculaire : $Tb.Sp = 1000/Tb.N - Tb.Th$ [29].

D'autres paramètres généraux ont été ajoutés depuis comme le volume de la moelle ($MaSV$) [34] ou encore la dimension fractale de l'os D [18, 42, 91].

Le facteur de forme de l'os trabéculaire $Tb.Pf$ est un index de connectivité basé sur la concavité et convexité de la surface trabéculaire totale. La concavité indique la connexité (noeuds) et la convexité indique des structures isolées, déconnectées. On le calcule en comparant la surface S et le volume V avant (S_1 et V_1) et après (S_2 et V_2) une dilatation d'un voxel : $Tb.Pf = (S_1 - S_2)/(V_1 - V_2)$ [43].

Des approches topologiques de l'analyse de l'architecture osseuse ont révélé de nouvelles informations intéressantes comme le nombre d'Euler (E/TV), qui donne un indice de connexion des objets [61], ou de nouvelles façons de calculer les précédentes (par exemple l'épaisseur trabéculaire [69]).

Avec l'apparition de nouvelles méthodes basées sur la squelettisation 2D [29, 30] et 3D [14, 15, 70], les informations structurelles comme le nombre de noeuds (NN^1), de points de terminaison (NF^1), leur ratio (N/F^1), l'index d'interconnexion (ICI) sont mesurables. Le ratio entre les noeuds et les points terminaux du réseau de trabécules est un index d'interconnectivité. Ces paramètres décrivent la relation entre le volume osseux et le réseau de trabécules.

Le tableau 6.2 présente les caractéristiques osseuses de l'analyse en 2 dimensions.

Paramètre	Abréviation	Unité
Volume Tissulaire	TV	mm^3
Volume osseux	BV	mm^3
Pourcentage de volume osseux	BV/TV	%
Surface tissulaire	TS	mm^2
Surface osseuse	BS	mm^2
Ratio surface/volume osseux	BS/BV	$1/mm$
Moyenne totale de l'aire du tissu transversal	T.Ar	mm^2
Moyenne totale du périmètre du tissu transversal	T.Pm	mm
Moyenne totale de l'aire de l'os transversal	B.Ar	mm^2
Moyenne totale du périmètre de l'os transversal	B.Pm	mm
Nombre moyen d'objets par image	Obj.N	-
Moyenne des aires des objets par image	Av.Obj.Ar	mm^2
Moyenne des diamètres des objets par image	Av.Obj.ECD	-
Moyenne des moments d'inertie polaires	MMI(polar)	mm^4
Moyenne d'excentricité	Ecc	-
Épaisseur transversale	Cs.th	mm
Épaisseur trabéculaire (modèle plat)	Tb.th(pl)	mm
Séparation trabéculaire (modèle plat)	Tb.Sp(pl)	mm
Nombre trabéculaire (plate model)	Tb.N(pl)	$1/mm$
Diamètre trabéculaire (modèle bâton)	Tb.Dm(rd)	mm
Séparation trabéculaire (modèle bâton)	Tb.Sp(rd)	mm
Moyenne des facteurs de forme	Tb.Pf	%
Centroid X	Crd.X	-
Centroid Y	Crd.Y	-
Centroid Z	Crd.Z	-
Moyenne des dimensions fractales	FD	-
Surface d'intersection totale	i.s	mm^2

TABLE 6.2 – Caractéristiques osseuses extraites lors de l'analyse 2D du volume d'images issu du microscanner.

Le tableau 6.3 présente les caractéristiques osseuses de l'analyse en 3 dimensions.

1. Abréviations en accord avec [29]

Paramètre	Abréviation	Unité
Volume Tissulaire	TV	mm^3
Volume osseux	BV	mm^3
Pourcentage de volume osseux	BV/TV	%
Surface tissulaire	TS	mm^2
Surface volumique	BS	mm^2
Surface d'intersection	i.s	mm^2
Ratio surface/volume osseux	BS/BV	$1/mm^2$
Densité de surface osseuse	BS/TV	$1/mm$
Facteur de forme trabéculaire	Tb.Pf	$1/mm$
Centroid X	Crd.X	mm
Centroid Y	Crd.Y	mm
Centroid Z	Crd.Z	mm
Indice de structure du modèle	SMI	-
Épaisseur trabéculaire	Tb.Th	mm
Nombre trabéculaire	Tb.N	$1/mm$
Séparation trabéculaire	Tb.Sp	mm
Degré d'anisotropie	DA	-
Valeur propre 1	-	-
Valeur propre 2	-	-
Valeur propre 3	-	-
Dimension fractale	FD	-
Nombre d'objet	Obj.N	-
Nombre de pores clôt	Po.N(cl)	-
Volume de pores clôt	Po.V(cl)	mm^3
Surface de pores clôt	Po.S(cl)	mm^2
Porosité close	Po(cl)	%
Volume d'espace poreux ouvert	Po.V(op)	mm^3
Porosité ouverte	Po(op)	%
Volume total d'espace poreux	Po.V(tot)	mm^3
Pourcentage de porosité totale	Po(tot)	%

TABLE 6.3 – Caractéristiques osseuses extraites lors de l'analyse 3D du volume d'images issu du microscanner.

Ces tableaux (6.2 et 6.3) présentent beaucoup d'informations osseuses à extraire. Ces paramètres sont extraits à partir du logiciel *CTan* du fabricant de microscanner *SkyScan*. La documentation des calculs des paramètres extraits est disponible sur le site du fabricant [87].

Seul le microscanner permet actuellement les mesures de ces paramètres. Le

scanner classique a une résolution insuffisante pour effectuer les mesures. De ce fait, ces mesures peuvent seulement se pratiquer sur un os biopsié. Cette technique étant invasive, elle n'est pas praticable dans une situation clinique. Car la taille d'analyse permise par le microscanner est trop petite pour imager le patient sans intervention. C'est pourquoi il est nécessaire de biopsier un petit morceau d'os pour pratiquer cet examen.

Un projet dirigé par Pierre Weiss appelé “Bioregos ” a été mis en place. Ce projet avait pour but de comprendre les mécanismes osseux et a l'ambition de proposer des solutions médicales associant des biomatériaux à des cellules vivantes du patient pour le traitement des pathologies ostéo-articulaires.

La deuxième partie du projet “Bioregos” dans laquelle s'intègre notre travail vise à caractériser l'architecture osseuse grâce à des coupes issues d'un microtomographe et parallèlement des images radiographiques standard. Les images de radiographies donneront des informations sur la texture osseuse alors que notre travail renseigne des informations sur l'architecture. Le but de ce travail sera de corréler les informations de texture et de structure osseuse afin de déterminer la qualité de l'os à partir de plusieurs images radiographiques. En d'autres termes, ce travail a pour but de créer une sorte de modèle de données de structure d'os de bonne qualité et de mauvaise qualité et de les associer à un modèle de texture osseuse. Ce qui permettra de disposer d'une correspondance de modèles entre les paramètres de la microarchitecture osseuse et ceux obtenus à partir de la texture.

6.3 Acquisition et pré-traitements

Les images que nous traitons sont issues d'un microtomographe Skyscan 1072 (“micro Computed-Tomography” noté μ -CT). Les images testées dans ce travail représentent une mandibule de chien avec une définition de taille de pixel de $19\mu m$.

Le volume μ -CT utilisé se compose de 1007 images $2D$ de 1024×1024 pixels. Il y a donc 1055916032 pixels dans ce volume μ -CT.

La figure 6.6 montre une représentation du volume d'images μ -CT obtenu avec le microtomographe.

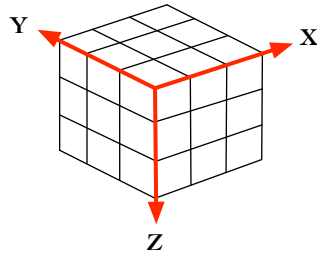
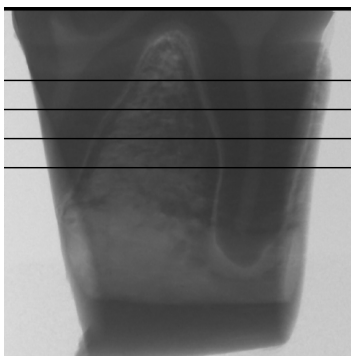


FIGURE 6.6 – Volume d'images 3D

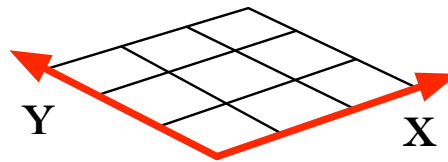
Afin d'extraire des informations de type 3D sur ce volume d'images avec notre méthode de squelettisation 2D nous découpons ce volume dans ces trois axes comme le représente la figure 6.12.

Ce découpage permet également de mieux se rendre compte de l'orientation de la formation osseuse. On sait que des propriétés mécaniques agissent sur la formation des os [32], par exemple dans la hanche l'os est structuré de telle façon qu'il sera capable de supporter le poids du corps et de répondre au mieux aux sollicitations mécaniques. Ces différents découpages du volume permettront de prendre en compte cette propriété mécanique.

Pour procéder à ce découpage il est important de bien comprendre de quoi il est composé. Les images correspondantes aux lignes représentées sur la figure 6.7 correspondent aux axes YX de la figure 6.6.



(a) Radiographie de la mandibule de chien



(b) axes YX

FIGURE 6.7 – Radiographie de la mandibule de chien.

Les images de la figure 6.8 sont situées dans le repère YX. Elles sont directement issues du microtomographe sans pré-traitements.

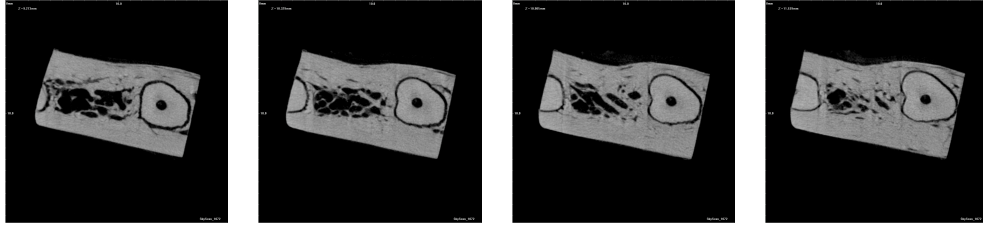


FIGURE 6.8 – Images μ -CT de l'os trabéculaire et de l'os cortical avec les deux racines de la dent visibles.

Un praticien (*eg* odontologue ici) sélectionne ensuite la zone d'intérêt avec un logiciel. Ces images segmentées sont représentées dans la figure 6.9.

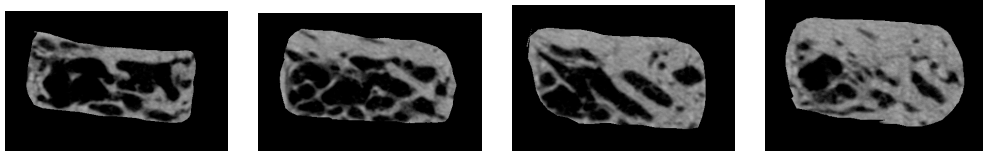


FIGURE 6.9 – Images μ -CT de la zone trabéculaire.

Avec ce même logiciel, le praticien applique une binarisation sur le volume d'images segmentées, la figure 6.10 représente ces images binarisées. Ce sont ces images que nous allons exploiter dans ce travail.

La figure 6.11 représente le volume d'image exploitable avec ses dimensions. Par cette figure nous savons que l'on a 137 images issues du plan YX (comme illustrées dans la figure 6.10) ayant pour dimension 224×418 pixels. On note qu'en réalité il y a 1007 images mais la zone trabéculaire est visible et exploitable sur seulement 137 images.

Afin d'extraire un maximum d'informations sur ce volume, nous appliquerons nos algorithmes sur les 3 directions du volume. Pour ce faire un découpage du volume est nécessaire. La figure 6.12 est une représentation des deux autres axes manquants.



FIGURE 6.10 – Images binarisées de la zone trabéculaire.

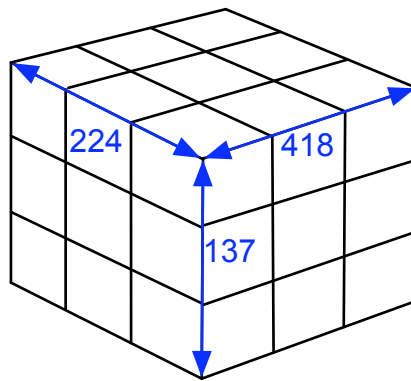
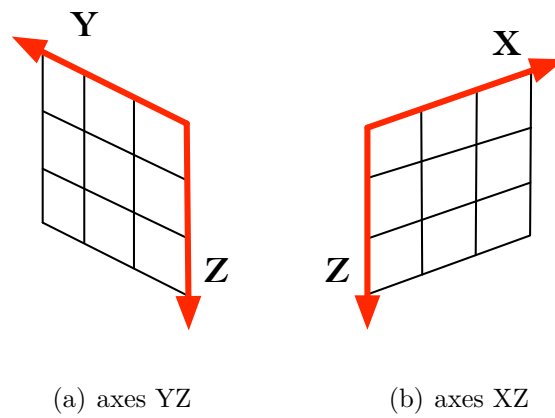


FIGURE 6.11 – Volume d'images binarisées de la zone trabéculaire avec ses dimensions.



(a) axes YZ

(b) axes XZ

FIGURE 6.12 – Découpage du volume d'image dans ces 3 axes.

Nous avons reconstitué les images dans l'axe YZ à partir des images binarisées par le médecin en sélectionnant toutes les x colonnes des 137 images du plan YX. De la même manière la construction des images dans le plan XZ se fait en sélectionnant les x lignes des 137 images du plan YX.

Ces images reconstituées sont présentées en figures 6.13 et 6.14.



FIGURE 6.13 – Images binarisées de la zone trabéculaire en XZ.



FIGURE 6.14 – Images binarisées de la zone trabéculaire en YZ.

Nous pouvons alors appliquer nos algorithmes de squelettisation sur ces données binaires.

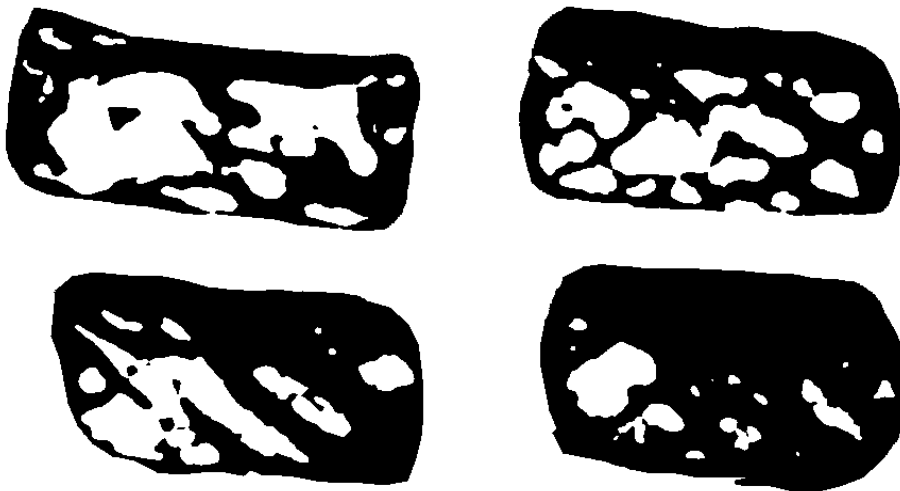


FIGURE 6.15 – Images binarisées de la zone trabéculaire.

La figure 6.16 présente les squelettes obtenus à partir des images binaires présentées dans la figure 6.15.

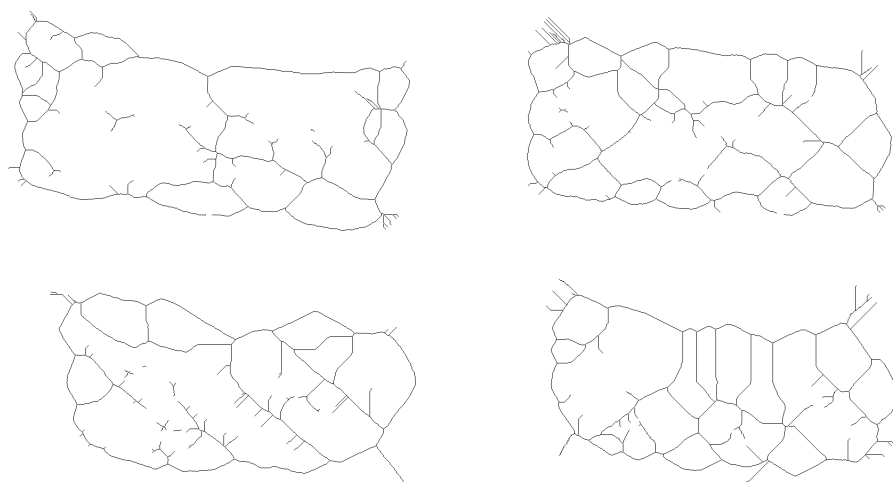


FIGURE 6.16 – Squelettes de la zone trabéculaire en YZ.

La figure 6.17 présente les squelettes obtenus à partir des images présentées dans la figure 6.13. Ces images et squelettes correspondent à la coupe XZ du volume d'images.

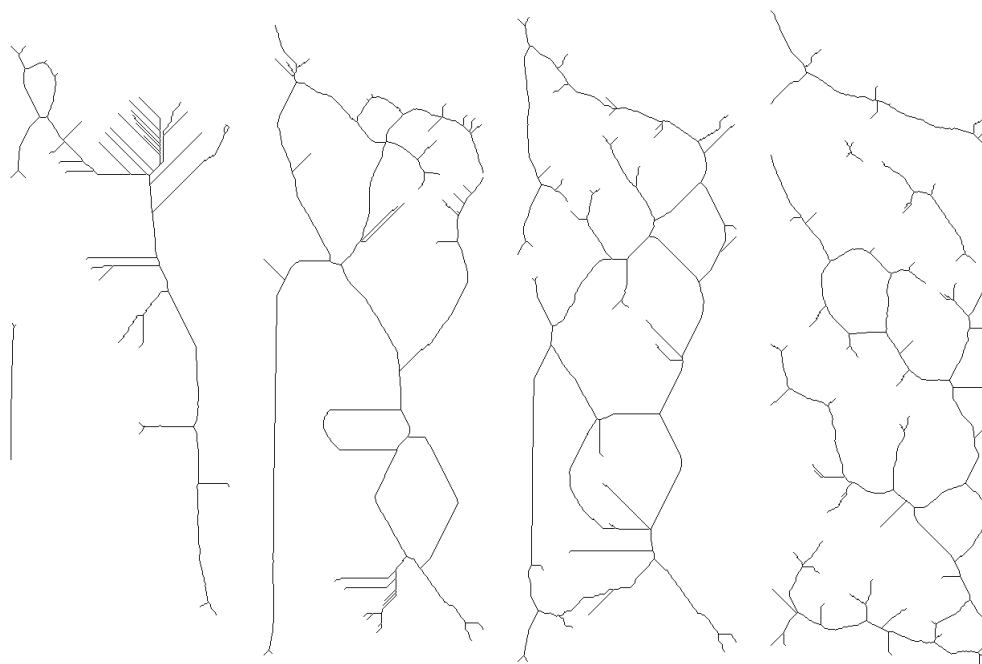


FIGURE 6.17 – Squelette de la zone trabéculaire en YX.

La figure 6.18 présente les squelettes obtenus à partir des images binaires présentées dans la figure 6.14. Ces images et squelettes correspondent à la coupe ZX du volume d'images.

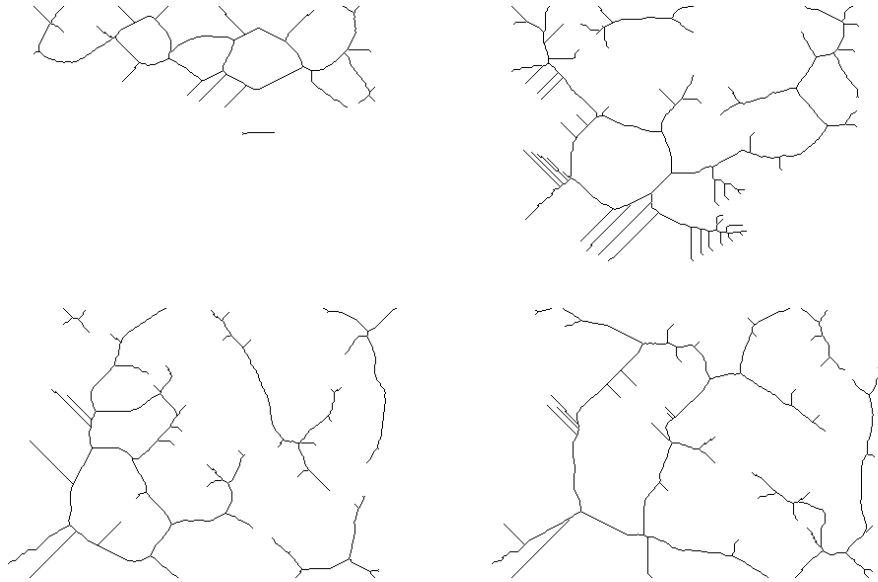


FIGURE 6.18 – Squelette de la zone trabéculaire en ZX.

C'est à partir de ces différents squelettes que nous allons extraire les caractéristiques osseuses dans le chapitre 7.

6.4 Conclusion

Afin d'étudier la micro architecture osseuse, nous avons d'abord identifié l'os trabéculaire. C'est une zone spongieuse qui se situe à l'intérieur de l'os cortical.

De nombreux paramètres de la littérature permettent de caractériser la micro architecture osseuse. Certaines de ces caractéristiques sont extraites à partir d'images en 2D et d'autres sur le volume d'images 3D. Dans notre cas, nous avons extrait les caractéristiques sur les squelettes 2D de la zone trabéculaire. Des post traitements sont appliqués pour segmenter la zone trabéculaire puis la binariser pour extraire son squelette. Afin d'obtenir un maximum d'informations sur la micro architecture osseuse à partir du volume d'images, nous avons découpé le volume en trois axes : YZ (format initial), YX et ZX.

Extraction des caractéristiques de l'os trabéculaire

Dans ce chapitre est décrite la façon dont les caractéristiques sont extraites. Suite à la partie II précédente, les informations recueillies sont :

- la carte de distance,
- les points de l'axe médian,
- le squelette (pas obligatoirement fin).

Dans la section 7.1 les définitions topologiques des caractéristiques sont présentées. Dans la section 7.2 la méthode d'extraction des caractéristiques de l'os est décrite. Enfin les résultats de cette extraction de caractéristiques seront présentés dans la section 7.3.

7.1 Définition des caractéristiques à extraire

Dans cette section nous allons redéfinir les caractéristiques à extraire, à partir du squelette calculé dans la partie précédente, avec un vocabulaire de morphologie mathématique. C'est-à-dire que l'on va analyser les points terminaux (7.1.1) et les points d'intersections (7.1.2) pixel par pixel et définir une fonction qui permettra de les extraire quelle que soit leur configuration dans l'image. De la même manière, nous allons analyser les informations de longueur (7.1.3) et de largeur (7.1.4) des trabécules.

Il faut garder à l'esprit que l'on va travailler sur un squelette, qui peut être considéré comme un graphe, et on va lui donner un sens médical. Pour cela, les trabécules analysées sont concrètement les arcs constituant le squelette. De ce fait une trabécule est comprise entre deux points spéciaux (nœuds ou terminaison).

7.1.1 Les points terminaux

On dit qu'un point est terminal s'il se trouve au bout d'une trabécule. De la même façon c'est un point qui ne possède qu'un seul voisin.

La figure 7.1 représente une forme rectangulaire et son squelette. Les points p et q sont des points terminaux suivant notre définition.

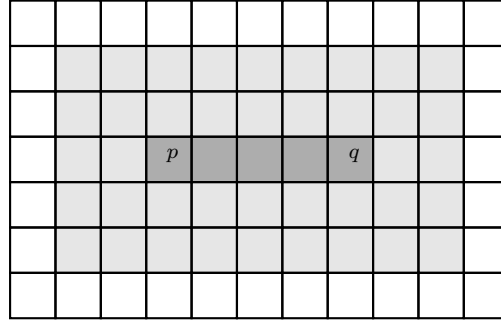


FIGURE 7.1 – Exemple de squelette et points terminaux.

Avant de détailler les propriétés d'un point terminal nous allons définir deux types de points terminaux. Nous avons d'une part les points terminaux identifiés sur le squelette calculé précédemment et d'autre part les points terminaux identifiés sur un squelette fin.

En effet, la méthode utilisée pour calculer le squelette ne garantit pas la finesse du squelette. La figure 7.2 représente une forme rectangulaire d'épaisseur paire avec son squelette.

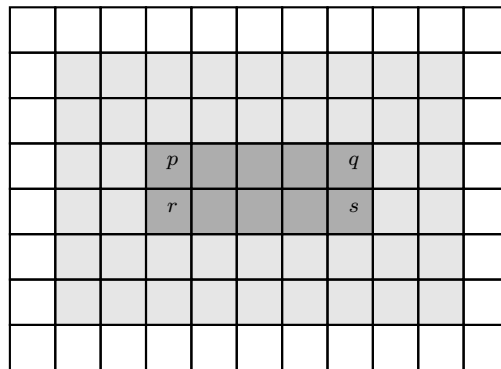


FIGURE 7.2 – Exemple de squelette et points terminaux.

Dans cet exemple, il n'y a pas de point qui ne possède qu'un seul voisin. Mais on peut noter que les points p , q , r et s peuvent être considérés comme des doubles points terminaux.

Définition 12 (Double points terminaux). *Un double point terminal est un point simple qui possède au plus 3 voisins.*

Algorithme 13: Identification des doubles points.

Données : Squelette non fin

Données : ligne l , colonne c

Résultat : oui ou non

```

1 nb ← nombre de voisins du point  $(l, c)$ ;
2 si  $nb \leq 3$  ET  $simple(l, c)$  alors
3   | retourne 1;
4 sinon
5   | retourne 0;
```

Dans cet algorithme, si la valeur retournée est égale à 1 alors le point de coordonnées l et c est un point terminal (ou double terminal suivant l'algorithme).

Dans ce travail nous allons d'avantage nous intéresser aux points terminaux détectés sur un squelette fin. La définition d'un point terminal sur un squelette fin est la suivante :

Définition 13 (Point terminal). *Un point terminal est un point simple qui ne possède qu'un seul voisin.*

Cependant il existe un cas particulier dans cette identification de points terminaux. La figure 7.3 montre ce cas particulier.

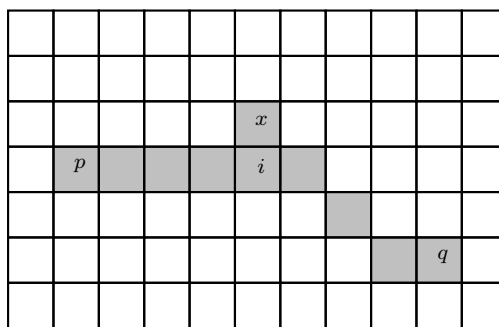


FIGURE 7.3 – Exemple de cas particulier lors de l'identification des points terminaux sur un squelette affiné selon la méthode décrite dans la sous section 5.4.2.

Dans ce cas particulier, les points p et q sont des points terminaux car ils n'ont qu'un seul voisin. Cependant le point x n'est pas considéré comme un point terminal car il a 3 voisins. Ce point x a pourtant des caractéristiques communes aux points p et q :

- il a au plus 3 voisins,
- il n'a qu'une seule composante connexe dans son voisinage.

En d'autres termes l'algorithme 13 de détection des doubles points terminaux sert également à identifier les points terminaux sur un squelette fin. Cependant ce cas particulier posera de nouveau problème. Ce problème sera détaillé dans la sous section 7.2.4.

7.1.2 Les points d'intersections

Le point d'intersection est un point de liaison de plusieurs trabécules. Sa définition 14 est donnée ci dessous.

Définition 14 (Point d'intersection). *Un point d'intersection est un point qui possède 3 ou 4 voisins et qui a également 3 ou 4 composantes 4-connexes dans son voisinage de type \mathcal{N}_2 .*

La figure 7.4 représente une forme rectangulaire avec un pixel supplémentaire au nord de la forme et son squelette. Le point identifié p est un point d'intersection car il a 3 voisins et 3 composantes connexes dans son voisinage.

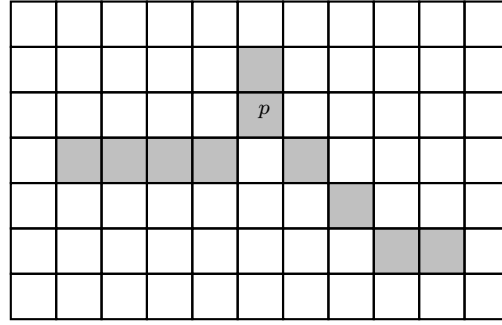


FIGURE 7.4 – Exemple de squelette et point d'intersection.

L'algorithme 14 de cette fonction est présenté ci dessous.

Algorithme 14: Identification du point d'intersection.

Données : Squelette fin

Données : ligne l , colonne c

Résultat : oui ou non

```

1 nb1 ← nombre de voisins du point  $(l, c)$ ;
2 nb2 ← Rutovitz du point  $(l, c)$ ;
3 si  $(nb1 = 3 \text{ OU } nb1 = 4) \text{ ET } (nb2 = 6 \text{ OU } nb2 = 8)$  alors
4   | retourne 1;
5 sinon
6   | retourne 0;
```

Dans cet algorithme, le point de coordonnées l et c est un point d'intersection si la valeur retournée est 1.

La fonction "Rutovitz", comme expliquée en section 1.3.3, calcule le "crossing number" du point courant. C'est à dire que l'on compte le nombre de changements de valeurs dans le voisinage du point de coordonnées l et c . Ainsi on compte le nombre de composantes connexes à ce point. Par exemple si la méthode retourne la valeur 6 cela veut dire qu'il y a 6 changements de valeur dans le voisinage du point et donc 3 composantes connexes dans son voisinage.

7.1.3 La longueur des trabécules

La longueur de chaque trabécule est une information importante à avoir. Elle permet notamment de caractériser partiellement la structure du réseau de trabé-

cules.

Pour mieux comprendre ce phénomène prenons l'exemple de la figure 7.5 et la définition de la longueur d'une trabécule 15.

Définition 15 (Longueur de trabécule). *La longueur d'une trabécule est le nombre de pixels situés entre 2 points spéciaux (points d'intersection ou/et point terminaux) dans le squelette fin.*

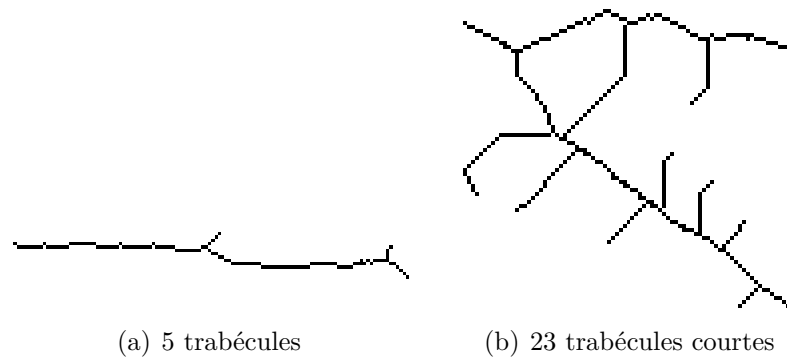


FIGURE 7.5 – Squelette de trabécules.

Ces deux images 7.5 sont extraites d'un seul squelette de trabécule mais il nous permet de nous rendre compte de l'importance de la longueur des trabécules dans la caractérisation de l'os. En adéquation avec l'information du nombre de noeuds, la longueur d'une trabécule permet de qualifier la structure osseuse de simple ou complexe.

Cette définition de la longueur permet également de ne pas prendre en compte certaines trabécules. Par exemple les trabécules de longueur 1 n'apportent pas d'informations pertinentes à l'analyse de la structure trabéculaire. Il sera donc facile de ne pas prendre en compte les trabécules ayant pour longueur un nombre inférieur à un certain seuil que les spécialistes pourront définir.

7.1.4 La largeur des trabécules

La largeur des trabécules est également une information très importante à prendre en considération. Par exemple, dans le processus de l'ostéoporose les trabécules s'amincissent considérablement ce qui les rend très fragiles. Il est vrai dans

ce cas que si l'on se réfère au squelette calculé, l'information de largeur est perdue. Le squelette étant fin il faudra se reporter à la carte de distance calculée précédemment. Pour chaque trabécule, pixel par pixel, les valeurs de la carte de distance

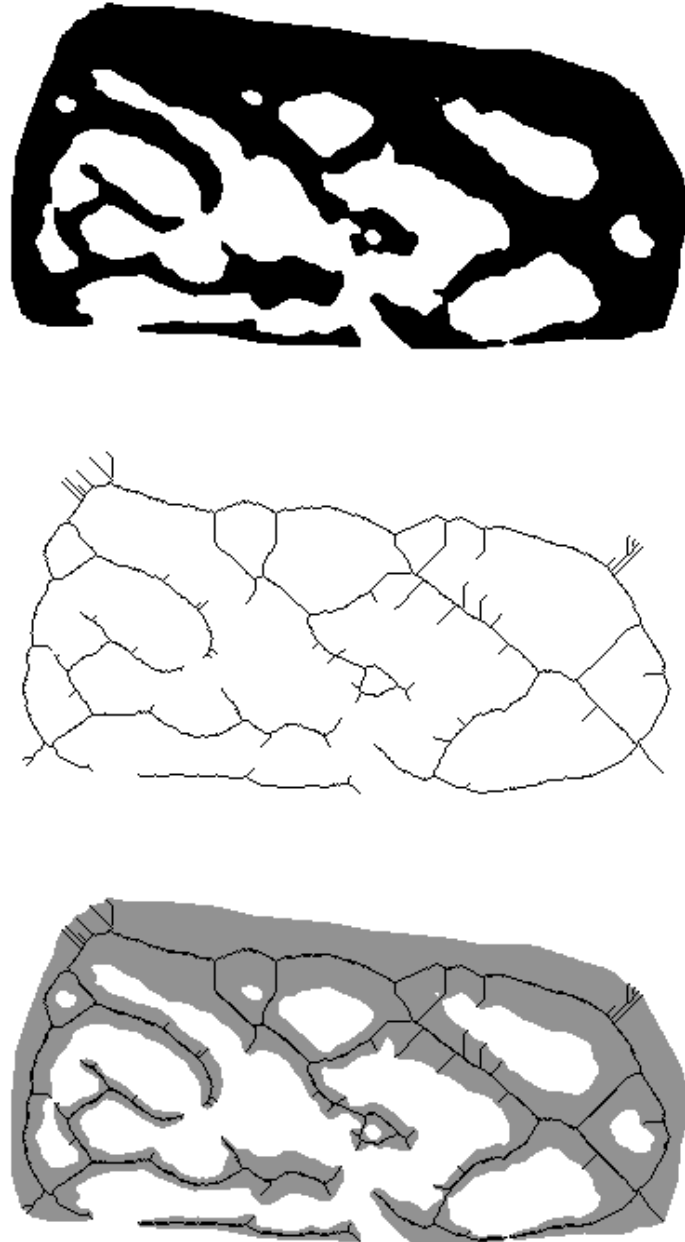
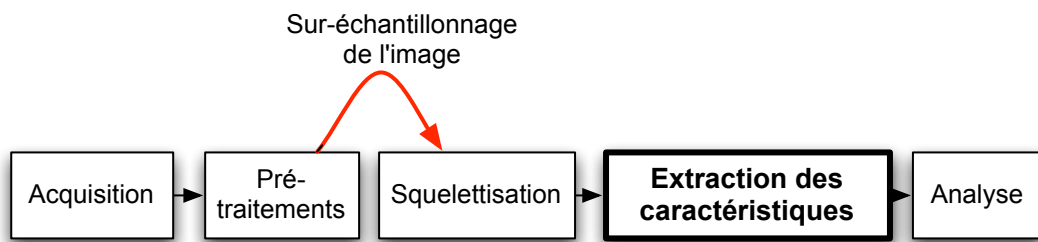


FIGURE 7.6 – Zone trabéculaire, son squelette et leur superposition.

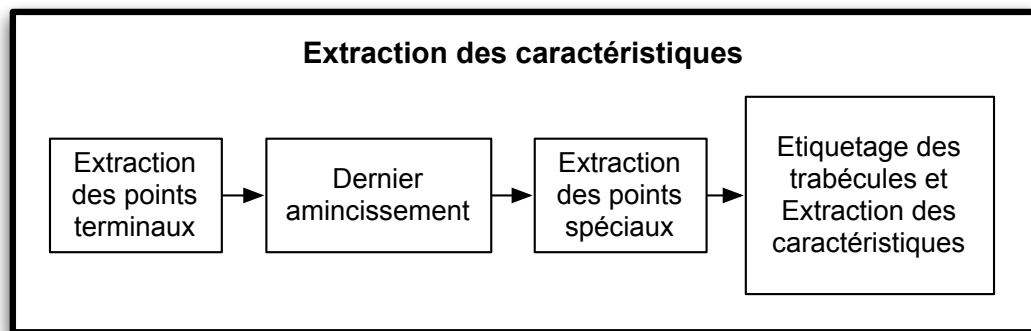
sont extraites et la moyenne de ces distances est calculée. La variance de cette largeur est également calculée afin de savoir si la largeur est uniforme sur toute la trabécule ou si il y a des écarts importants de largeur sur une même trabécule.

7.2 Méthode d'extraction des informations

Les caractéristiques osseuses étant définies dans la section précédente, il s'agit ici de présenter et de détailler l'appel de ces fonctions d'extraction d'informations. Pour cela, la figure 7.7 représente les différentes étapes du processus d'extraction des informations en partant de l'image radiographique de l'os segmenté et binarisé jusqu'à l'obtention des caractéristiques.



(a) Schéma général



(b) Schéma d'extraction des caractéristiques

FIGURE 7.7 – Schéma des différentes étapes du processus menant à l'extraction des caractéristiques osseuses.

7.2.1 Rappel des étapes précédentes

À ce niveau du travail nous avons 3 volumes d'images binaires correspondants aux images des axes XY, XZ et YZ. On rappelle les propriétés des 3 volumes :

- Volume XY : 137 images de 224x418 pixels,
- Volume YZ : 418 images de 137x224 pixels,
- Volume XZ : 224 images de 418x137 pixels.

Ces 3 volumes d'images sont ensuite sur-échantillonnés (section 5.4.1) de telle sorte que chaque image possède une résolution supérieure. Une fois les images redimensionnées, en un seul balayage de l'image la carte de distance de cette image est calculée. Dans cette partie d'extraction des caractéristiques osseuses seule la distance d_4 a été utilisée. Il est possible d'extraire les informations sur un squelette calculé à partir d'une carte de distance d_8 ou d_{ns} , cependant dans la suite de chapitre nous noterons les problèmes liés à ces deux distances. Une fois la carte de distance calculée, une identification des points de l'axe médian est opérée. Ces points seront considérés comme points d'ancrage du squelette lors de la suppression des points simples. Le squelette ainsi calculé en un balayage de l'image est presque fin. Comme vu précédemment (5.4.1), le fait d'augmenter la résolution de l'image permet d'extraire un squelette plus fin notamment lorsque l'épaisseur de la forme est paire. Toutefois il reste des configurations où ce traitement ne permet pas d'affiner le résultat. C'est le cas lorsque le squelette, avec une épaisseur de 2 pixels, a une direction diagonale.

Ce dernier cas étant gênant pour l'extraction des caractéristiques, il est donc nécessaire d'affiner le squelette une fois de plus.

7.2.2 Dernier amincissement

Si un dernier balayage de l'image est appliqué en supprimant tous les points simples (algorithme naïf 4), des morceaux de trabécules (voire des trabécules entières) disparaîtront. C'est à dire qu'un point terminal d'une trabécule est forcément simple donc supprimé. Ce dernier point n'existant plus son voisin subit le même sort jusqu'à changement de la topologie de la forme.

La figure 7.8 illustre ce phénomène sur un exemple de squelette. Les valeurs de la figure 7.8(a) représentent l'ordre dans lequel les pixels deviennent simples suivant

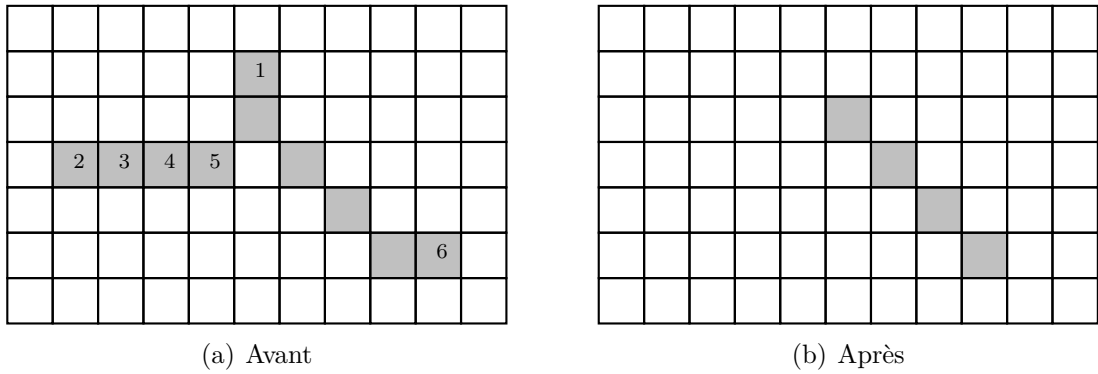


FIGURE 7.8 – Exemple d'un dernier amincissement du squelette.

le balayage. La figure 7.8(b) représente ce squelette après le dernier amincissement. Ce résultat montre deux problèmes :

- Il y a suppression de deux trabécules,
- La suppression n'est pas symétrique (car avec un seul balayage elle ne peut pas être symétrique).

Le principe de ce dernier amincissement n'est pas de supprimer des trabécules mais de les affiner. C'est pourquoi une détection des points terminaux de chaque trabécule est faite afin de les considérer comme points d'ancrage au moment de la squelettisation au lieu de considérer les points de l'axe médian. La figure 7.9(a) représente un squelette non fin et ses points terminaux. La première étape consiste à supprimer un point terminal (le premier rencontré) collé à un autre point terminal (double points terminaux). La seconde étape est de procéder à un dernier amincissement afin d'obtenir un squelette fin sans risque de supprimer des trabécules comme le montre la figure 7.9(b).

7.2.3 Extraction des points spéciaux

Une fois le squelette affiné, une identification des points spéciaux (points terminaux et d'intersections) est appliquée. Pour les enregistrer et les localiser, une image temporaire de ces points spéciaux est créée. La figure 7.10 représente l'image de ces points spéciaux, le squelette dans cette image est présenté ici dans le but de mieux se rendre compte de la justesse de la détection des points spéciaux.

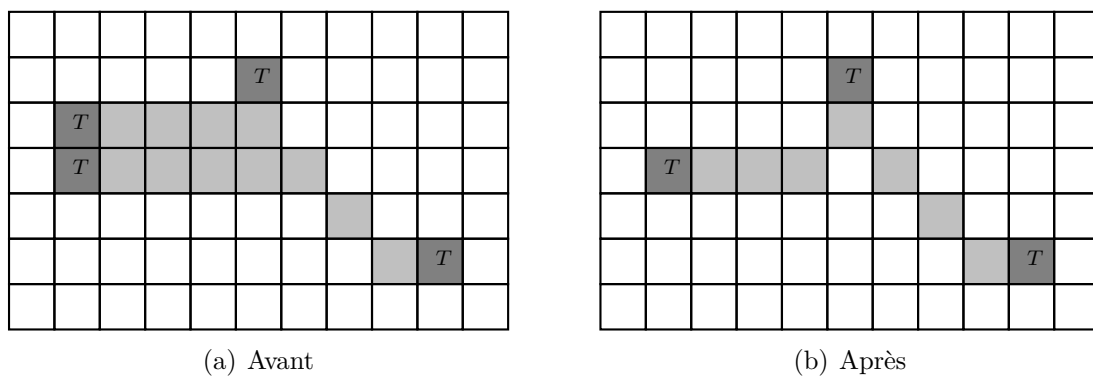


FIGURE 7.9 – Exemple d'un dernier amincissement du squelette avec points d'ancrage.

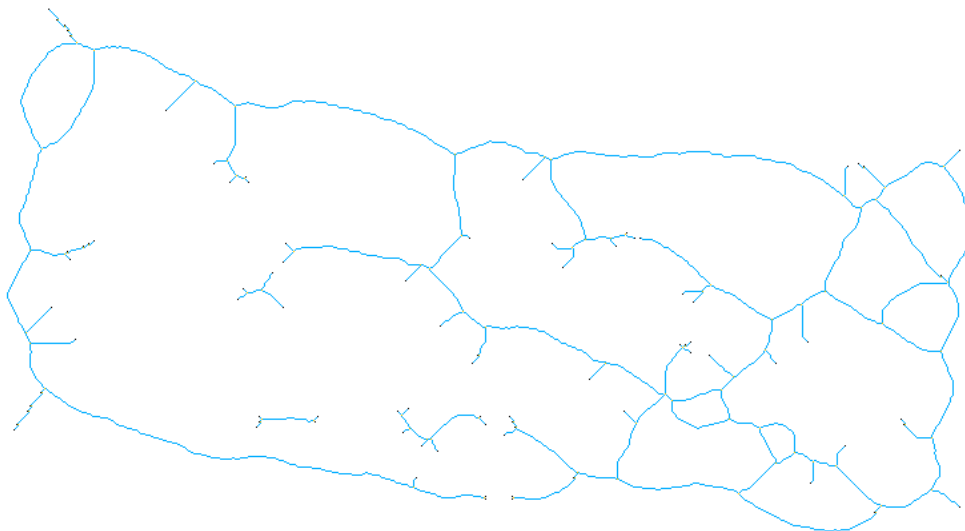


FIGURE 7.10 – Squelette d'os avec les points spéciaux.

La figure 7.11 présente la différence entre le squelette de la figure 7.10 avant (points gris) et après (points noirs) le dernier amincissement. Seules ces deux zones de l'image diffèrent car la squelettisation sur une image sur-échantillonnée produit globalement un squelette fin.



FIGURE 7.11 – Différence entre les squelettes avant et après le dernier amincissement.

La figure 7.12 représente un morceau agrandi de la figure 7.10. Les points jaunes représentent les points d'intersection quant aux points noirs ils représentent les points terminaux des trabécules.

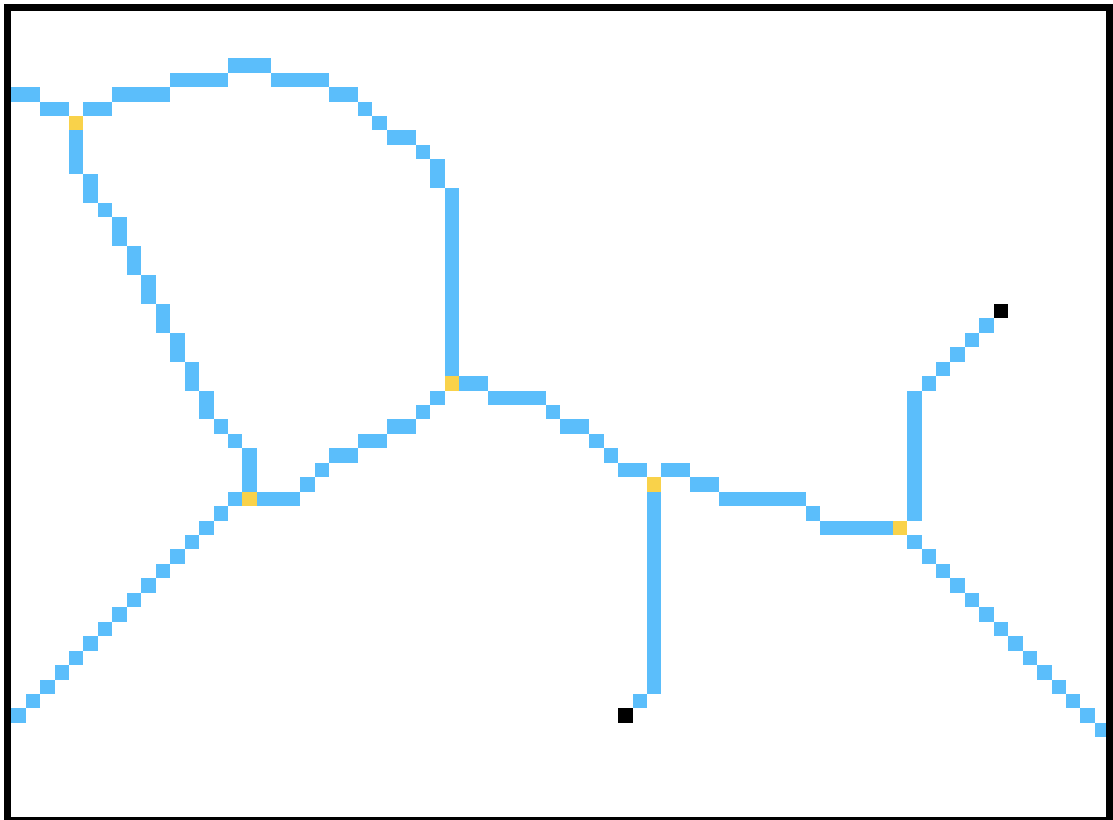


FIGURE 7.12 – Zoom de la figure 7.10

7.2.4 Étiquetage des trabécules

L'étiquetage est une fonction (voir la section sur les composantes connexes 1.3.1) qui permet de donner une étiquette différente aux composantes connexes d'une image.

De la même manière ici, une soustraction des points d'intersections avec le squelette est appliquée ce qui revient à déconnecter chaque trabécule pour les rendre non connexes entre elles. La fonction d'étiquetage attribuera de cette manière une étiquette à chaque trabécule ce qui les identifiera.

La figure 7.13 représente les trabécules étiquetées où chaque couleur correspond à une étiquette. La couleur du fond a volontairement été corrigée en noir afin de mieux se rendre compte des différentes couleurs présentes dans l'image.

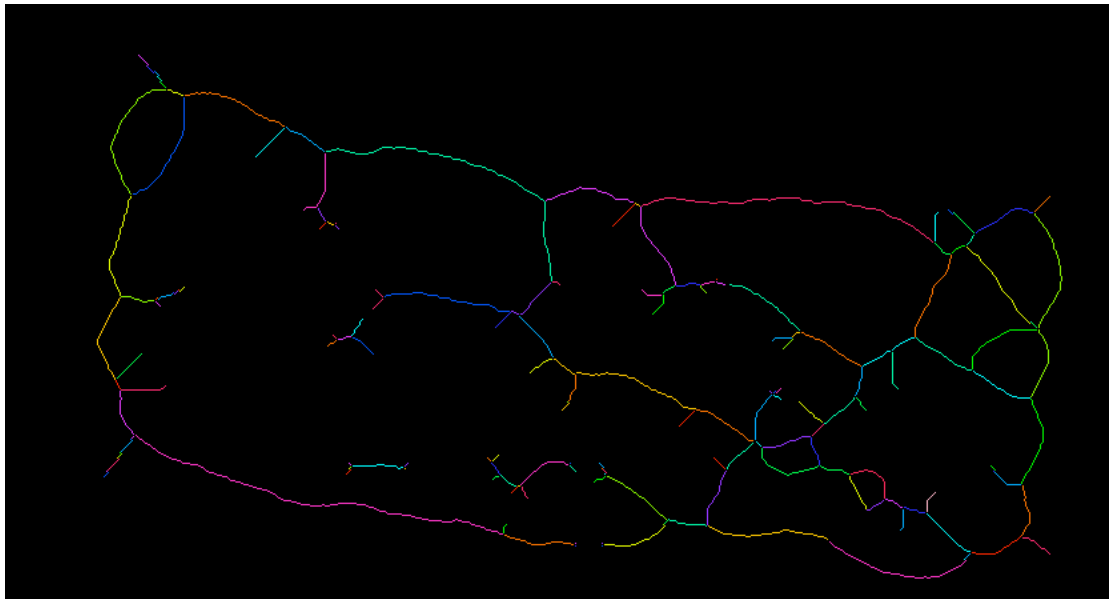


FIGURE 7.13 – Trabécules étiquetées.

La figure 7.14 représente un morceau agrandi de la figure 7.13. On notera que les points d'intersections ont été retirés du squelette et que de ce fait les trabécules sont déconnectées les unes des autres.

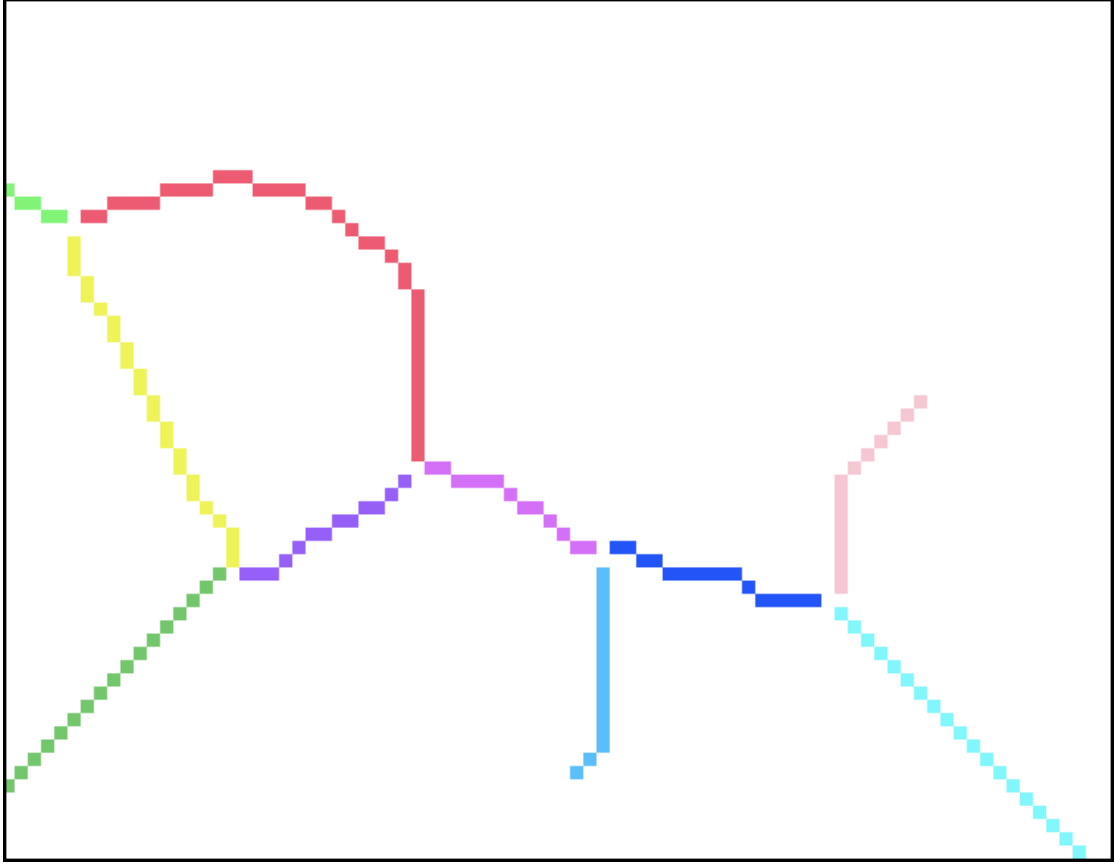


FIGURE 7.14 – Zoom de la figure 7.13

Rappelons que ces traitements sont opérés sur des squelettes calculés à partir de la carte de distance d_4 . Cette méthode est également applicable aux squelettes calculés à partir des cartes de distance d_8 et d_{ns} . Cependant dans ces derniers cas les squelettes obtenus génèrent des imperfections dans la détection des trabécules. Pour comprendre pourquoi, il faut remonter à l'étape d'identification de l'axe médian. Les points de l'axe médian sur les cartes de distances d_4 , d_8 et d_{ns} ne sont pas les mêmes car les disques de distances sont différents. À l'étape de squelettisation, les points simples sont supprimés sauf les points de l'axe médian. À ce stade, les squelettes sont différents car ils dépendent de l'axe médian.

La figure 7.15 présente des morceaux de squelettes d'os trabéculaire calculés avec les distances d_4 , d_8 et d_{oct} après le dernier amincissement. Le problème rencontré est plus flagrant dans le cas de la distance d_8 . Sur ce résultat on peut constater

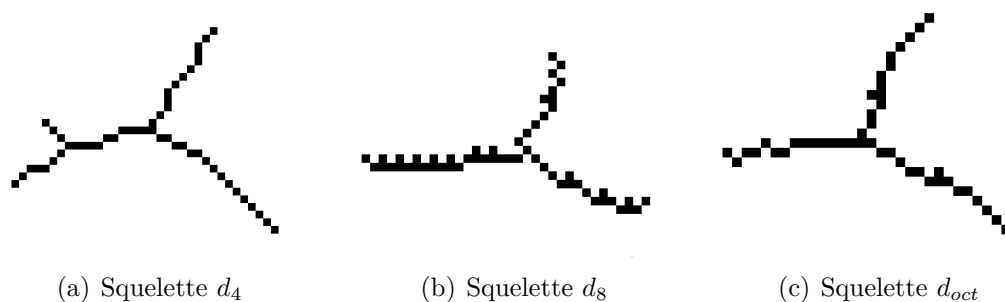


FIGURE 7.15 – Morceaux de squelettes d_4 , d_8 et d_{oct} .

qu'une des trabécules a une forme de "peigne". Ces points superflus constituant les dents du "peigne" sont des points de l'axe médian et c'est pourquoi ils ne sont pas supprimés. Ces points ne sont pas supprimés lors du dernier amincissement car ils ont d'abord été détectés comme points terminaux.

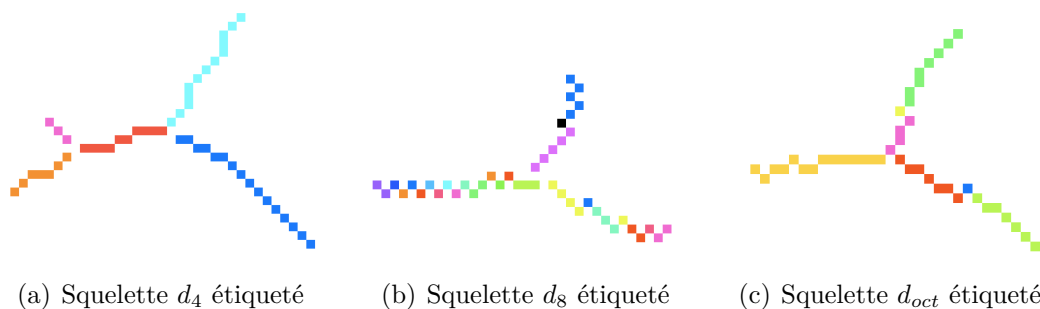


FIGURE 7.16 – Morceaux de squelettes d_4 , d_8 et d_{oct} étiquetés.

La figure 7.16 présente l'étiquetage des squelettes de la figure 7.15. On se rend compte grâce aux différentes couleurs que la détection des trabécules est imparfaite dans les cas d_8 et d_{oct} car les trabécules sont coupées à différents endroits. Pour remédier à ce problème il suffirait d'ajouter un pré-traitement situé avant l'étape du dernier amincissement qui consisterait à supprimer les points terminaux ayant pour voisin un point d'intersection.

Dans la pratique, une structure de données "trabecule" est créée pour chaque trabecule trouvée où leur identifiant sera le numéro de l'étiquette attribuée à l'ensemble des pixels que composent la trabecule. Cette structure de données comporte plusieurs champs :

- ID,
- Longueur,
- Largeur,
- Ecart type Largeur,
- Nb pt term,
- Nb pt inter,
- Nb pt SP,
- Nb trab voisine.

L'entier *ID* est l'identifiant de la trabécule donné par la fonction d'étiquetage. Cet *ID* permet d'accéder directement à chaque trabécule grâce à une boucle sur cet entier.

La variable *Longueur* correspond à la longueur de la trabécule qui est en fait le nombre de pixels que composent celle ci. Comme expliqué dans la section 7.1.3, cette information est importante à extraire car elle nous renseigne sur la fragilité de l'ossature.

La fragilité de l'ossature est également mesurée grâce à la variable *Largeur*. Cette variable peut être assimilée au paramètre *Tb.Th* du tableau 6.1. La section 7.1.4 met en évidence l'importance de l'épaisseur des trabécules dans la définition de la qualité osseuse. La variable *Largeur* contient la moyenne de la largeur de la trabécule. Cette largeur (équation 7.1) est calculée grâce à la somme des valeurs de la carte de distance observées à chaque point de la trabécule divisée par le nombre de points (*Longueur*).

$$Largeur = \frac{1}{nb} \sum_{p=1}^{nb} DT_X(p) \quad (7.1)$$

Mais cette largeur moyenne n'est pas suffisante pour qualifier une trabécule. Car il suffit qu'une trabécule ne soit pas uniformément épaisse pour que cette variable *Largeur* n'ai pas de sens. C'est pourquoi nous lui associons son écart type 7.2.

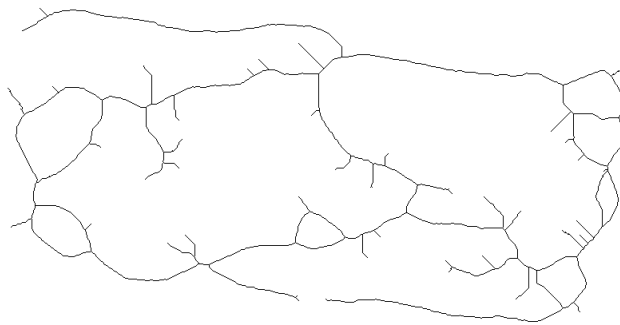
$$\text{ÉcarttypeLargeur} = \sqrt{\frac{1}{nb} \sum_{p=1}^{nb} DT_X(p)^2 - Largeur^2} \quad (7.2)$$

où la variable *Largeur* correspond à la largeur moyenne de la trabécule calculée avec l'équation 7.1.

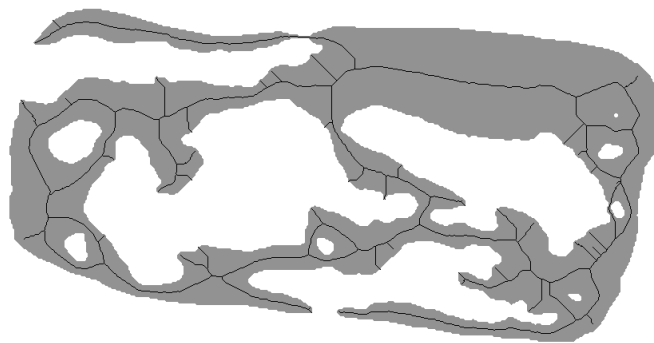
Pour illustrer l'intérêt de l'écart type, la figure 7.17 présente un exemple d'image d'os trabéculaire binarisée, le squelette associé et la superposition de l'os et du squelette.



(a) Os trabéculaire



(b) Squelette de l'os



(c) Squelette et os

FIGURE 7.17 – Exemple d'image d'os trabéculaire binarisée et son squelette.

<i>Largeur*</i>	<i>Écart type Largeur</i>	<i>Largeur</i>	<i>Écart type Largeur*</i>
22,00	0,00	7,76	0,42
23,00	0,00	25,15	0,43
23,00	0,00	9,25	0,43
23,00	0,00	9,25	0,43
23,00	0,00	10,25	0,43
24,00	0,00	11,25	0,43
24,00	0,00	11,25	0,43
24,00	0,00	14,25	0,43
24,00	0,00	17,25	0,43
24,00	0,00	24,25	0,43
24,00	0,00	24,25	0,43
24,00	0,00	5,14	0,43
24,00	0,00	4,27	0,45
24,00	0,00	5,70	0,46
24,00	0,00	6,30	0,46
24,00	0,00	6,33	0,47
24,00	0,00	6,33	0,47
24,00	0,00	6,67	0,47
24,00	0,00	7,67	0,47
24,25	0,43	9,33	0,47
24,25	0,43	11,67	0,47
24,33	0,47	14,33	0,47
24,50	0,50	15,33	0,47
24,50	0,50	24,33	0,47
24,57	0,49	31,67	0,47
25,00	0,00	38,33	0,47
25,00	0,00	29,64	0,48
25,00	0,00	8,63	0,48
25,00	0,00	33,63	0,48
25,00	0,00	31,40	0,49
25,00	0,00	11,43	0,49
25,00	0,00	17,43	0,49
25,07	5,11	24,57	0,49
25,15	0,43	3,44	0,50
26,14	0,35	3,50	0,50
27,00	0,00	5,50	0,50
27,00	0,00	7,50	0,50

TABLE 7.1 – Échantillon de caractéristiques observé sur l'os 7.17.

Le tableau 7.1 présente une partie des résultats (largeur et écart type) de l'os présenté sur la figure 7.17. La totalité de ces résultats (300 trabécules) est représentée en annexe 2.1.

Lorsque l'écart type est élevé (par exemple 5, 11 dans le tableau) cela signifie que les largeurs observées en chaque point sont différentes. Du point de vue de la structure, ce type de résultat indique que la trabécule grossit (ou s'amincit) continuellement (entre les deux points spéciaux).

Les variables *NB pt term* et *NB pt inter* correspondent respectivement au nombre de points terminaux et d'intersections dans la trabécule. La variable *NB pt SP* correspond au nombre de points spéciaux de la trabécule. Cette variable doit toujours avoir la valeur 2 car c'est l'addition du nombre de points terminaux et d'intersections d'une trabécule. Ces variables sont extraites par le logiciel d'odontologie sous le paramètre *N/T* du tableau 6.1

La variable *NB trab voisine* correspond au nombre de trabécules voisines à la trabécule en question. Dans la pratique il s'agit d'étudier le voisinage \mathcal{N}_2 des points d'intersection et d'ajouter 1 dans le champ *NB trab voisine* des trabécules voisines identifiables par leur étiquette.

7.3 Résultats

Le programme conçu pour extraire les caractéristiques de l'os trabéculaire écrit et enregistre un fichier texte contenant ces informations. Ces données peuvent donc être analysées post-traitement.

Nous mettons en évidence 3 types de trabécules :

- Type 1 : trabécule ayant 2 points d'intersections,
- Type 2 : trabécule ayant 1 point d'intersection et 1 point terminal,
- Type 3 : trabécule ayant 2 points terminaux (trabécule isolée).

La figure 7.18 représente l'histogramme du nombre de trabécules en fonction de leur type dans les trois volumes. Cet histogramme met en évidence le fait qu'il y a plus de points spéciaux dans le volume YZ. Ceci s'explique par le fait que dans

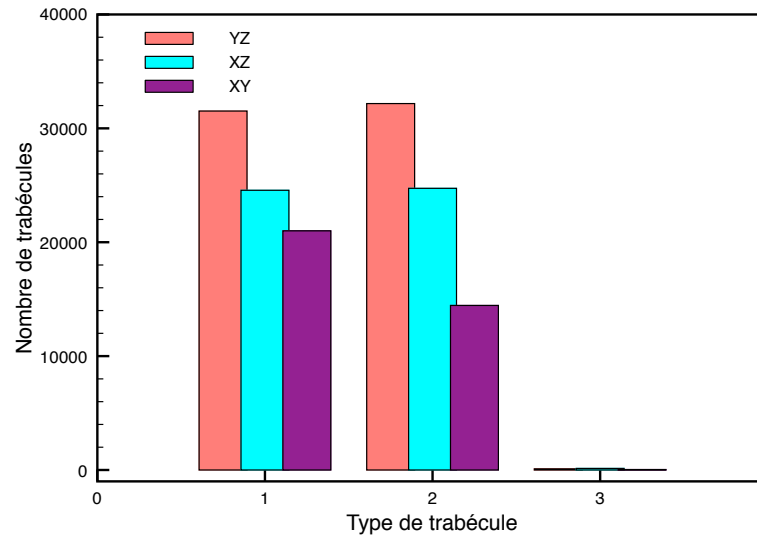


FIGURE 7.18 – Histogramme du nombre de points terminaux par trabécules dans les 3 volumes.

ce volume il y a plus d'images à analyser et donc plus de trabécules à extraire.

On rappelle les propriétés des 3 volumes :

- Volume XY : 137 images de 224x418 pixels,
- Volume YZ : 418 images de 137x224 pixels,
- Volume XZ : 224 images de 418x137 pixels.

Plus précisément, la figure 7.18 souligne le fait que, quel que soit le volume d'images étudié, il y a très peu de trabécules ayant 2 points terminaux. Ce qui veut dire que dans les 3 volumes d'images il y a très peu de trabécules isolées. Et inversement l'histogramme montre qu'il y a beaucoup plus de trabécules de type 1 que de trabécules de type 3 (isolées) ce qui met donc en évidence une forte interconnectivité du réseau trabéculaire.

La figure 7.19 représente 3 nuages de points correspondants à la longueur moyenne en fonction de la largeur moyenne de chaque trabécules dans les 3 volumes d'images.

Ce nuage de points est homogène.

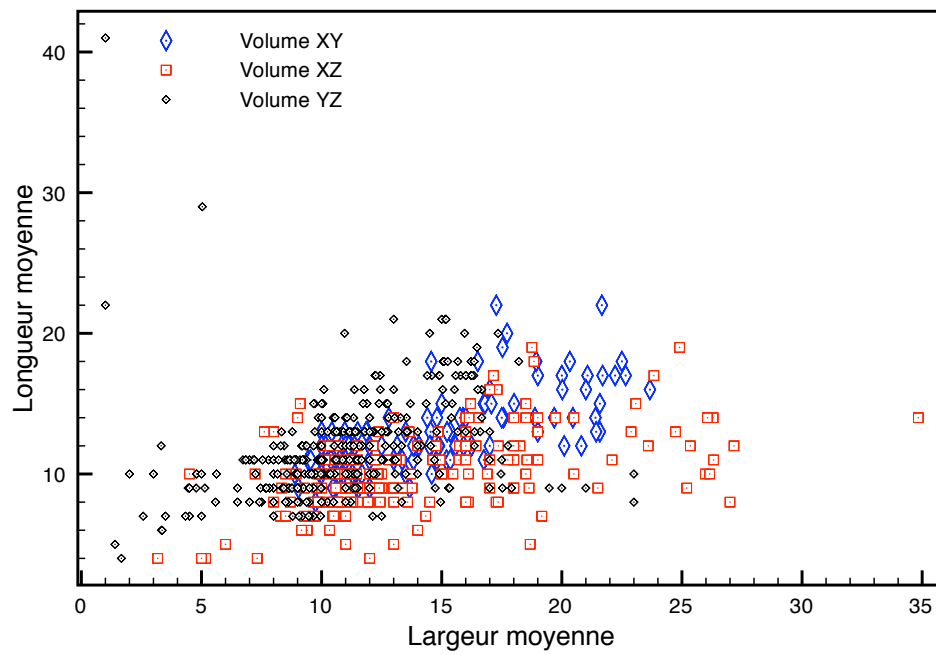


FIGURE 7.19 – Nuage de points représentant la longueur moyenne en fonction de la largeur moyenne de chaque trabécule dans les 3 volumes.

La grande majorité des trabécules ont une longueur moyenne inférieure à 20 pixels. En ce qui concerne la largeur médiane on peut noter 2 tendances en fonction du volume dans lequel les trabécules sont analysées :

- Volume XY : largeur minimale de 4 pixels et largeur maximale de 24 pixels,
- Volume XZ : largeur minimale de 3 pixels et largeur maximale de 35 pixels,
- Volume YZ : largeur minimale de 1 pixel et largeur maximale de 23 pixels.

Les trabécules analysées dans le volume XZ ont des largeurs plus variables que celles des trabécules analysées dans les volumes XY et YZ.

Conclusion

Dans cette partie est introduite une application à l'algorithme de squelettisation en un balayage. Cette application consiste à extraire les caractéristiques osseuses de l'os afin de déterminer sa qualité.

En collaboration avec le Professeur Yves Amouriq, du Laboratoire d'Ingénierie Ostéo-Articulaire et Dentaire "LIOAD", nous avons identifié les caractéristiques osseuses importantes à étudier. Au sein du laboratoire, il utilise un logiciel permettant de caractériser la structure osseuse. Ce logiciel fournit une quantité de caractéristiques osseuses grâce à une analyse 2D et 3D des images. Dans cette partie, une analyse en 2 dimensions du volume d'images a été réalisée dans le domaine discret. C'est pourquoi les caractéristiques extraites sont des informations discrètes :

- Longueur trabéculaire,
- Épaisseur trabéculaire,
- Nombre de noeuds du réseau trabéculaire,
- Nombre de point terminaux du réseau trabéculaire.

Nous avons également ajouté une autre variable, l'écart type de l'épaisseur trabéculaire, donnant une information de dispersion de la largeur sur la trabécule. Cet ensemble d'informations permet de qualifier la structure osseuse sur un plan 2D. Dans le but d'analyser le volume d'images, l'analyse en 2 dimensions n'est pas suffisante. C'est pourquoi le découpage du volume en 3 axes a été pratiqué. Ainsi l'extraction des caractéristiques osseuses a été faite sur 3 axes et l'analyse de la structure osseuse se pratique en corrélant les résultats des 3 axes.

Cependant, cette technique d'extraction d'informations de la structure osseuse présente des limites. La principale, et pas des moindres, reste la façon dont la région d'intérêt est sélectionnée. Le squelette étant très sensible au contour de l'objet à étudier, la sélection joue un rôle primordial. Des barbules apparaissent lorsque la sélection n'est pas lisse et de ce fait nous analysons ces barbules comme des trabécules.

Conclusion générale

Entouré et protégé de l'os cortical, l'os trabéculaire est une partie spongieuse de l'ensemble osseux. Son étude contient deux aspects : la texture et la structure. Dans ce travail, nous nous sommes attachés à étudier l'aspect structurel de l'os trabéculaire. À partir d'un volume d'images issues d'un microscanner, nous avons développé des algorithmes permettant d'extraire des caractéristiques structurales de l'os.

Pour ce faire, nous avons choisi d'utiliser une méthode bien connue dans la littérature en reconnaissance de formes : la squelettisation.

Cette méthode permet d'extraire le squelette d'une forme binaire, à savoir un sous ensemble de la forme, restructurable, fin, centré et homotope à la forme. Parmi les méthodes existantes, notre choix s'est porté sur une méthode de squelettisation menée par la distance. Quelque soit la méthode choisie, ces algorithmes nécessitent plusieurs balayages de l'image pour calculer le squelette.

La contribution majeure de ce travail de thèse est d'extraire un squelette avec un seul balayage de l'image.

Dans un premier temps nous avons calculé une carte de distance en un balayage de l'image. Une translation des centres des disques a été effectuée pour obtenir une carte de distance "asymétrique" : DT'_X . Les distances ainsi calculées sont d_4 , d_8 et d_{ns} . Nous avons également présenté un algorithme permettant de calculer DT_X à partir de DT'_X .

La seconde étape a consisté à extraire les points de l'axe médian à partir de DT'_X . Ces points servent de points d'ancrage au squelette ce qui lui permettra d'être centré dans la forme.

Dans la dernière étape dite de squelettisation, nous avons procédé à un amincissement de la forme guidé par la distance. C'est-à-dire que nous avons supprimé les points simples sans supprimer les points de l'axe médian. Pour effectuer ce traitement en un seul balayage de l'image, nous avons étudié la dépendance entre pixels et avons formulé un ordre partiel. La démarche a été de modifier l'algorithme de référence rlk pas à pas (algorithme intermédiaire $l'rk'$) afin d'obtenir un algorithme en un balayage de l'image $l'k'r$. Le respect de l'ordre partiel, précédemment défini, assure l'identité des résultats lors des changements effectués entre chacun

des 3 algorithmes proposés.

Notre méthode permet d'obtenir un squelette :

- Homotope car la suppression des points est régie par une fonction de simplicité,
- Centré car l'axe médian est conservé,
- Reconstructible car l'information de distance est sauvée en mémoire,

Cependant le squelette calculé n'est pas spécialement fin. Effectivement dès lors que l'on souhaite conserver l'axe médian dans le squelette celui ci ne peut pas être toujours fin. Nous avons apporté des solutions qui permettent de le rendre fin (dernier amincissement ou sur-échantillonnage de la grille de travail) mais ces post-traitements décentrent légèrement le squelette.

Ce nouvel algorithme de squelettisation calculé grâce à un balayage de l'image offre un avantage indéniable : il permet de calculer un squelette à la volée. Plusieurs aspects de cet algorithme sont appréciables. L'architecture séquentielle de cet algorithme permet une implantation facile aux circuits de logique programmables (FPGA, INTEL,...). Cet algorithme est moins gourmand en ressources mémoire car les accès à la mémoire sont mieux localisés que ceux d'un autre algorithme. L'extraction du squelette en un balayage de l'image offre des opportunités non égalées quant au traitement des images de très grande taille. Par exemple, le traitement d'images en cours d'acquisition donc non bornées temporellement (images obtenues par des caméras linéaires par exemple).

Dans le but d'analyser l'architecture osseuse, nous avons appliqué notre algorithme à un volume d'images 2D issue d'un microscanner. À partir du squelette de la zone trabéculaire, nous avons extrait des informations :

- Le nombre de trabécules présentes dans l'image,
- La longueur de chaque trabécule,
- La largeur (et écart type associé) de chaque trabécule,
- Le nombre de noeuds de chaque trabécule,
- Le nombre de points terminaux de chaque trabécule.

Notre algorithme, permettant une analyse 2D des images, extrait des informations sur un volume tronqué (plusieurs couches d'images) et ignore l'information

se trouvant en profondeur. De ce fait, les données recueillies ne sont pas directement représentatives de la structure trabéculaire dans son ensemble. C'est pour cette raison que nous avons découpé ce volume d'images selon 3 axes (le premier étant naturellement donné par le bloc d'images). L'analyse osseuse a donc été réalisée en corrélant les résultats obtenus sur les 3 axes.

La caractérisation de l'architecture osseuse n'est pas la seule application à laquelle ces nouveaux outils peuvent être utilisés. D'autres domaines d'application nécessitent des outils de reconnaissance de formes auxquels nos traitements peuvent être ajoutés. De nombreux logiciels de traitement d'image incluent des traitements de reconnaissance de formes et donc peuvent utiliser ces nouveaux algorithmes.

Perspectives

Plusieurs aspects de ce travail n'ont pas été développés à ce jour. Il s'agit ici de faire l'état des lieux de plusieurs pistes qui pourront être suivies afin de peaufiner ce travail.

Plus généralement, ce travail s'est attaché à extraire des caractéristiques sur l'architecture osseuse mais il est également intéressant, dans le cadre de l'étude osseuse, d'analyser la texture osseuse. Ce travail a déjà débuté dans le cadre de la thèse de Pauline Bléry au sein du LIOAD. De son côté, elle analyse la texture osseuse. Ce travail, combiné au notre, permettra de mettre en corrélation des informations osseuses afin de juger la qualité osseuse d'un patient. Ces deux travaux sont complémentaires car dans un cas (le notre) les traitements demandent une imagerie invasive (avec biopsie de l'os) ils nous permettent de donner une sorte de modèle de données de structure d'os de bonne qualité et de mauvaise qualité. Alors que dans l'autre cas (celui de Pauline Bléry), le modèle de type texture osseuse est obtenu grâce à une radiographie.

Ces deux types de modèles permettront d'établir une correspondance afin d'établir un diagnostic sans avoir recours à une méthode invasive.

Cette section présente plusieurs pistes à notre travail. On peut classer ces améliorations en fonction du temps de leur mise en œuvre (à court, moyen ou long terme) :

- Transposition de l'algorithme 2D vers la 3D (section 1) : long terme,
- Changement de distances (section 2) : moyen terme,
- Amélioration de la méthode d'extraction des caractéristiques osseuses (section 3 : court terme,
- Utilisation de la méthode pour une autre application (section 4).

1 Squelettisation 3D en un balayage

La squelettisation présentée est une méthode applicable au domaine 2D. Cependant cette méthode n'est pas facilement transposable au domaine 3D pour différentes raisons.

Premièrement, le nombre de cas particuliers traité en 2D risque d'être multiplié si

l'on ajoute une dimension à notre domaine de travail. Deuxièmement, la fonction de simplicité d'un point est complètement différente du 2D au 3D. Ce qui suggère une reformulation de cette fonction et une réadaptation de celle ci à la démarche "translatée".

2 Distances utilisées

Dans ce travail, la squelettisation est menée en distance. Les distances utilisées sont d_4 , d_8 et d_{ns} (d_{oct} pour les exemples). Si ces trois distances sont présentées et développées dans ce travail, ça ne veut pas dire qu'une autre distance ne serait pas envisageable. Par exemple la distance de voisinage proposée par Robin Strand et Benedek Nagy [85] peut être modifiée pour être calculable en un balayage de l'image.

3 Amélioration de la méthode d'extraction des caractéristiques osseuses

Le squelette peut être vu comme un graphe et donc être traité comme tel.

Différents types de connexions peuvent être identifiés qui renseignent sur la nature du graphe. L'idée serait d'avoir un masque de pondération (comme celui de la figure 7.20) et de l'appliquer au voisinage.

	x	1
8	4	2

FIGURE 7.20 – Masque de pondération des types de connexions.

Ainsi, suivant la valeur de retour de ce masque, le pixel x peut être identifié en fonction de sa connexion avec ses voisins.

Les questions auxquelles répondrait cette fonction de connexion seraient :

- “Est ce que les points p et q sont voisins au sens du graphe (dit vg) ?”
- “Combien le point p a t'il de vg ?”

Si p n'a qu'un seul vg alors p est un point terminal. En revanche si p a plus de 2 vg , alors p est un point d'intersection. Avec cette méthode il ne serait plus nécessaire de supprimer les points d'intersection pour pouvoir appliquer la fonction d'étiquetage sur le squelette. Cette fonction de connexion identifie également le sens de connexion. Ce sens de connexion indique de quelle façon le pixel est relié à son vg . La connexion du point avec un de ses voisins (non vg) ne compte pas car on risquerait de créer des boucles de connexion de qui rendrait notre graphe de connexion non topologiquement équivalent au squelette.

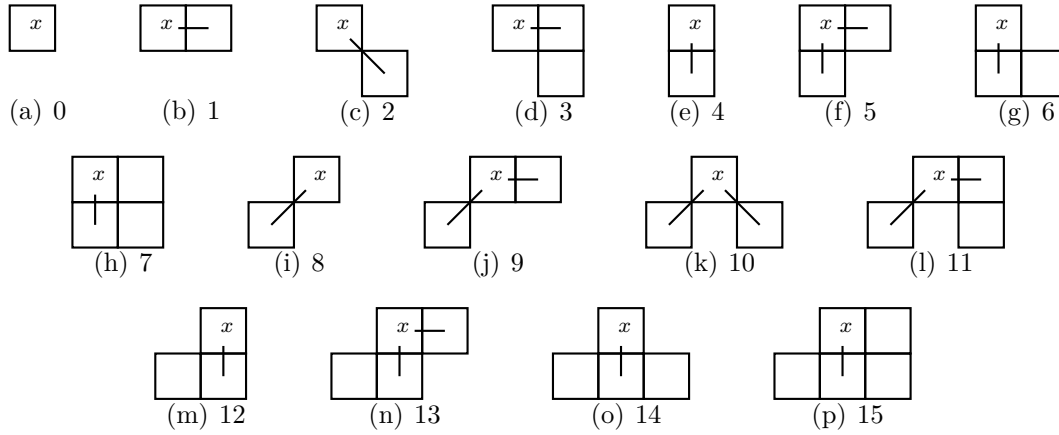


FIGURE 7.21 – 16 premières configurations de connexions.

Pour mettre en pratique cette idée il est nécessaire de créer un tableau qui recense les types de connexions du graphe. Les 16 premières cases de ce tableau sont réservées au 16 premières connexions représentées dans la figure 7.21. À partir de celles ci on remplit les 239 autres. Cela fait un total de 256 cases correspondant aux 256 configurations du voisinage possibles.

Dans ce tableau doivent figurer les informations suivantes :

- Le nombre de vg ,
- Le sens de connexion.

Par exemple pour le cas correspondant à la configuration 15 de la figure 7.21 on peut dire que le point x a 1 “ vg ” et que le sens de connexion est 6 au sens de Freeman. Alors que pour le cas correspondant à la configuration 10 de la figure 7.21 on peut dire que le point x a 2 “ vg ” et que les sens de connexion sont 5 et 7

au sens de Freeman.

Initialement, le code de Freeman [39] est utilisé pour coder les contours d'un polygone dans un plan discret. Il suffit alors d'enregistrer chaque déplacement (arête du polygone) grâce au code des directions présenté dans la figure 7.22.

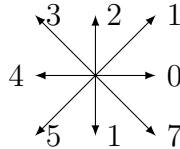


FIGURE 7.22 – Code de Freeman.

4 Autres applications

Dans ce travail, notre algorithme en un balayage de l'image nous sert à extraire des informations sur l'architecture osseuse. Néanmoins, la squelettisation est un outil présent dans différents domaines. Le fait que notre algorithme se déroule sur un balayage unique de l'image et que la totalité de l'image n'est pas nécessaire au calcul du squelette peut apporter des avantages. Par exemple, cet algorithme est adapté aux traitements à la volée des images issues d'une caméra linéaire.

Quatrième partie

Annexe

Cette annexe a pour but de présenter un complément d'informations et de résultats sur ce travail de thèse.

Dans un premier temps, des graphiques comparant les temps d'exécution des algorithmes de référence et de nos algorithmes sont présentés.

Dans un second temps, toutes les caractéristiques osseuses extraites par les praticiens du laboratoire LIOAD sont présentées sur un volume d'images.

1 Temps d'exécution

Plusieurs algorithmes ont été présentés dans ce travail :

- calcul des cartes de distances asymétriques DT'_X (d_4 , d_8 et d_{oct}),
- calcul de l'axe médian pour ces distances asymétriques MA'_X ,
- calcul du squelette.

1.1 La base d'images testées

Nos algorithmes ont tous été comparés à une base d'image composée de 57 images binaires de taille différente et contenant des formes différentes. Pour mieux apprécier cette diversité, la figure 1.1 présente une courbe du nombre de pixels de la forme et du nombre total de pixels de la forme de toutes ces images.

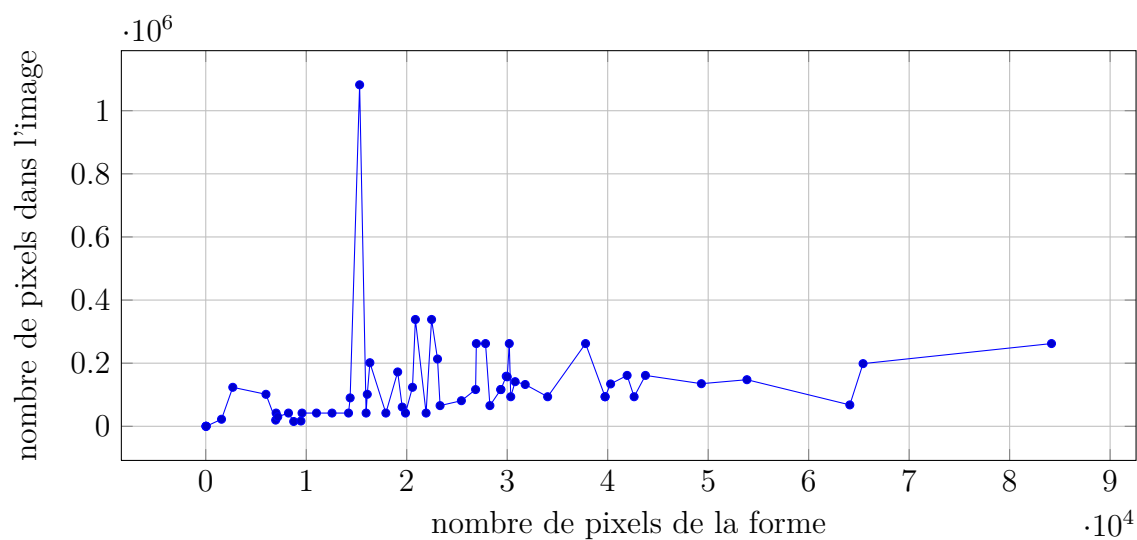


FIGURE 7.23 – Nombre de pixels total de l'image en fonction du nombre de pixels de la forme.

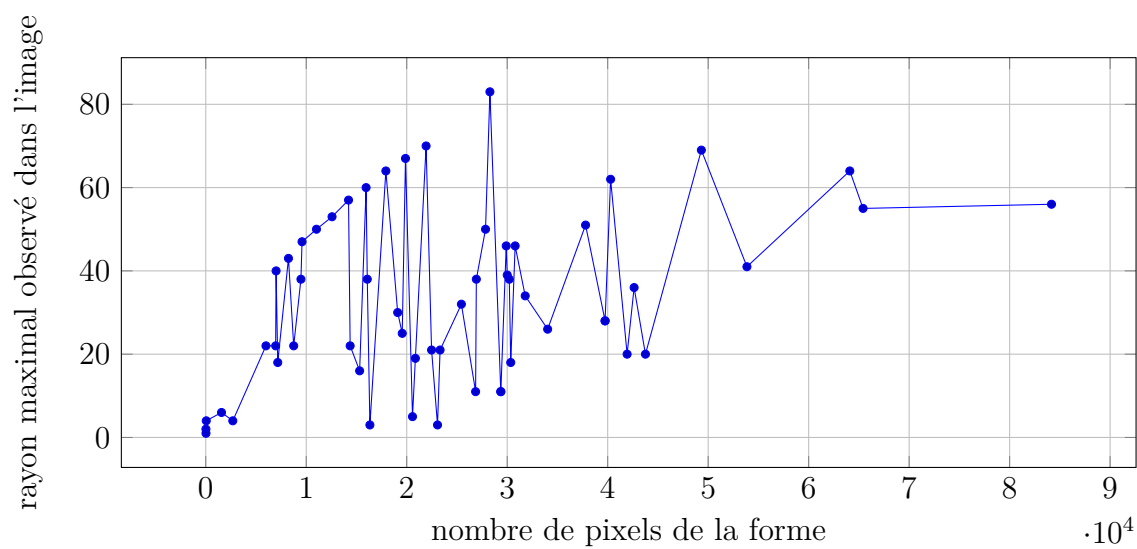


FIGURE 7.24 – Rayon maximal observé dans l'image en fonction du nombre de pixels dans la forme.

1.2 Carte de distance

La figure 1.2 présente le graphique du nombre de pixels total de la forme en fonction du rayon maximal de la forme observé dans l'image sur toute la base d'images.

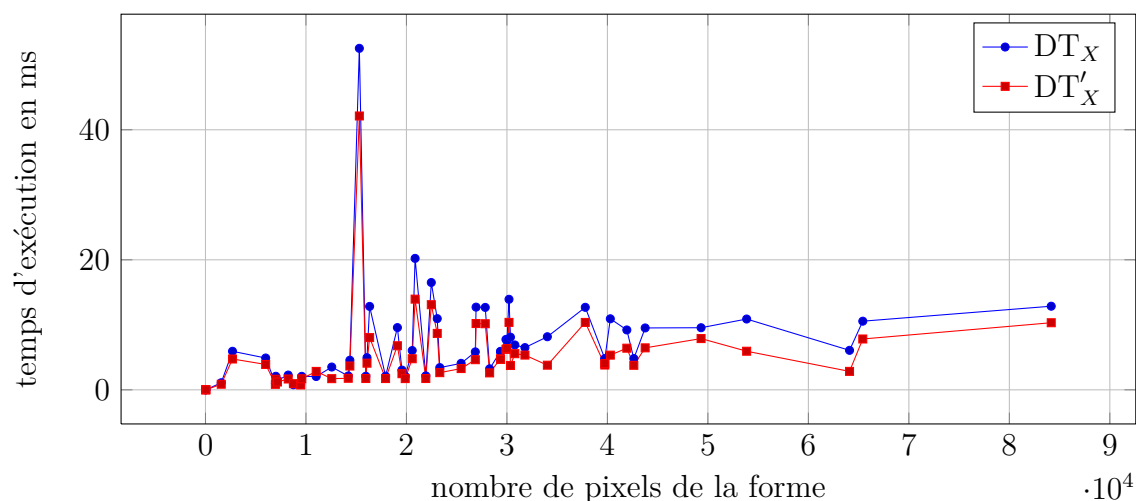


FIGURE 7.25 – Temps de calcul de DT_X et de DT'_X en d_4 en fonction du nombre de pixels de la forme.

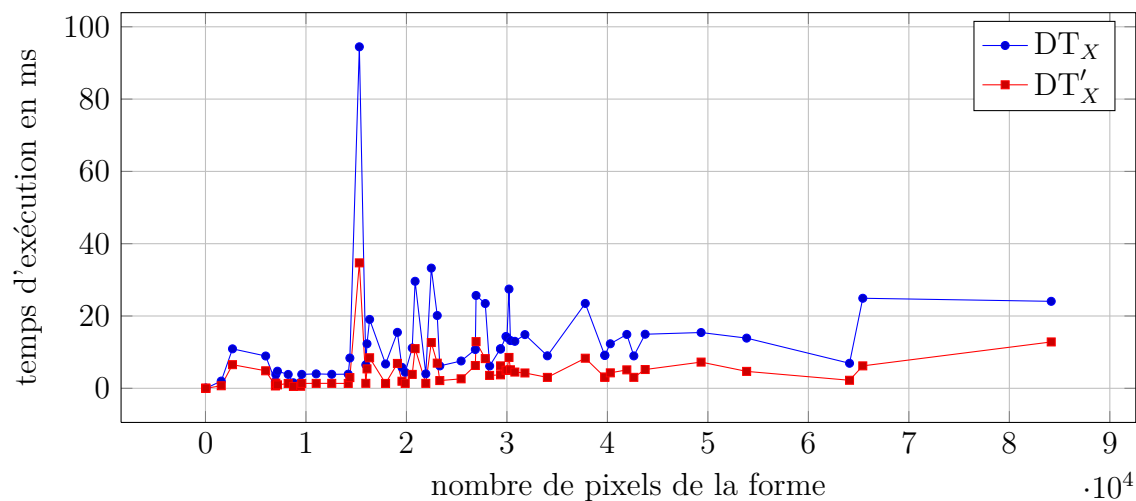


FIGURE 7.26 – Temps de calcul de DT_X et de DT'_X en d_8 en fonction du nombre de pixels de la forme.

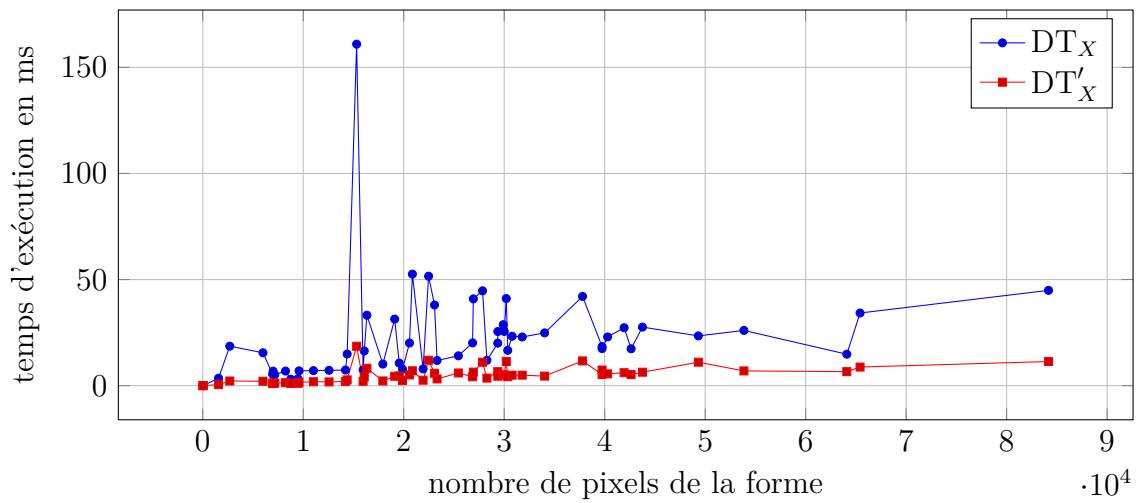


FIGURE 7.27 – Temps de calcul de DT_X et de DT'_X en d_{oct} en fonction du nombre de pixels de la forme.

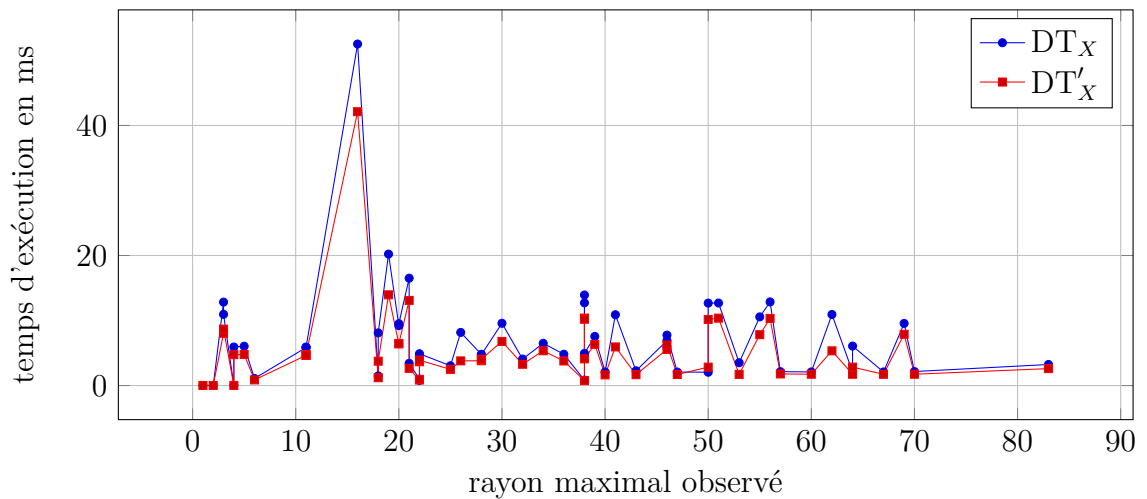


FIGURE 7.28 – Temps de calcul de DT_X et de DT'_X en d_4 en fonction du rayon maximal observé dans la forme.

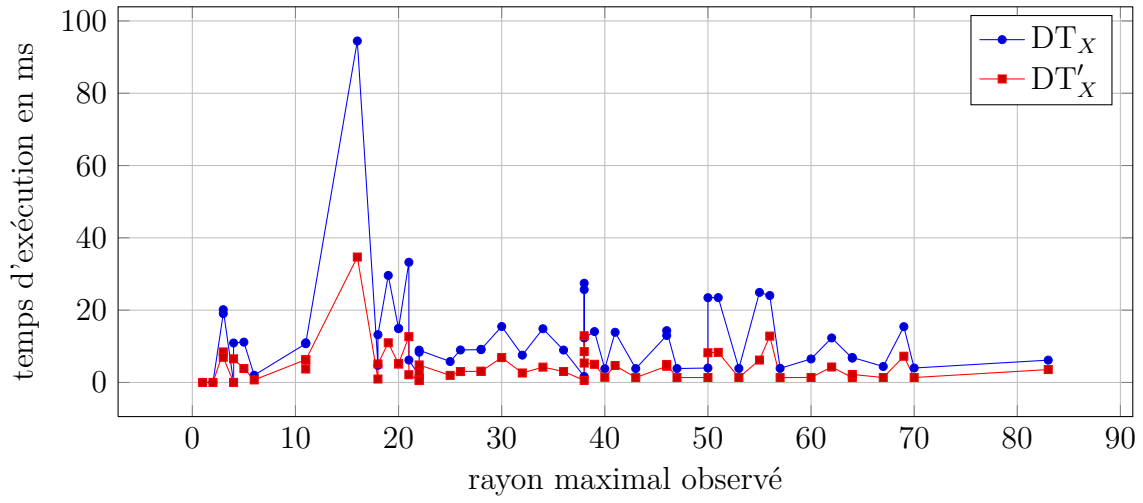


FIGURE 7.29 – Temps de calcul de DT_X et de DT'_X en d_8 en fonction du rayon maximal observé dans la forme.

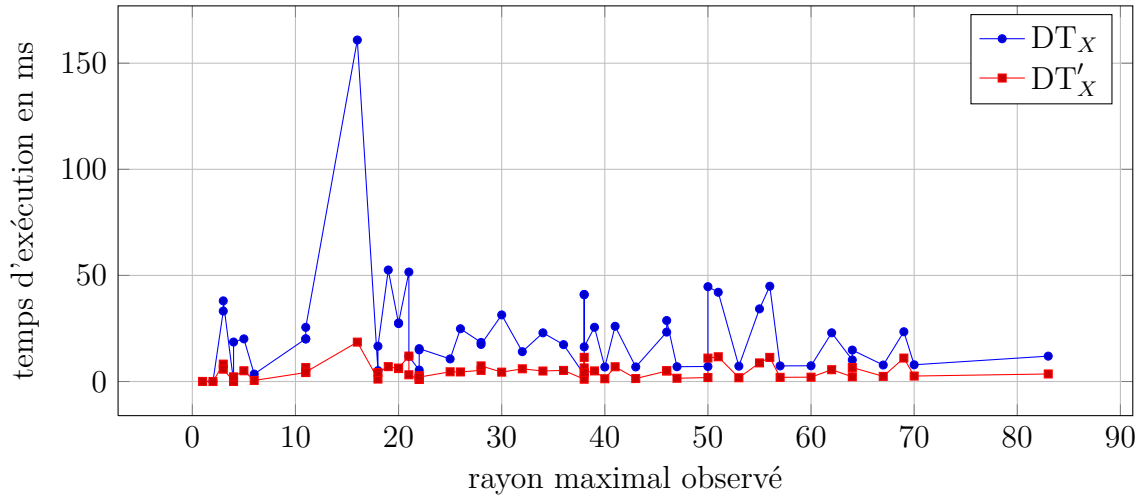


FIGURE 7.30 – Temps de calcul de DT_X et de DT'_X en d_{oct} en fonction du rayon maximal observé dans la forme.

1.3 Axe médian

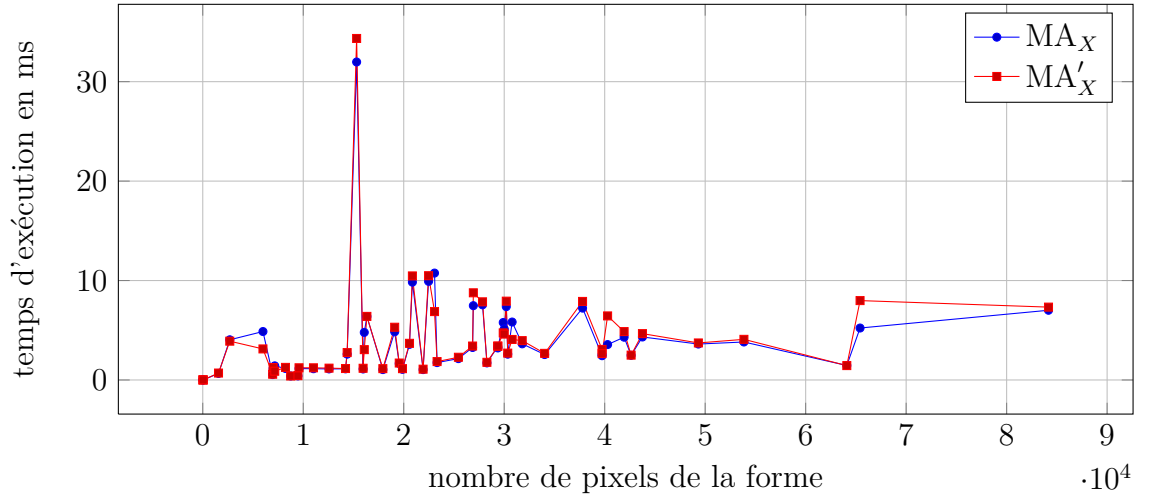


FIGURE 7.31 – Temps de calcul de MA_X et de MA'_X en d_4 en fonction du nombre de pixels de la forme.

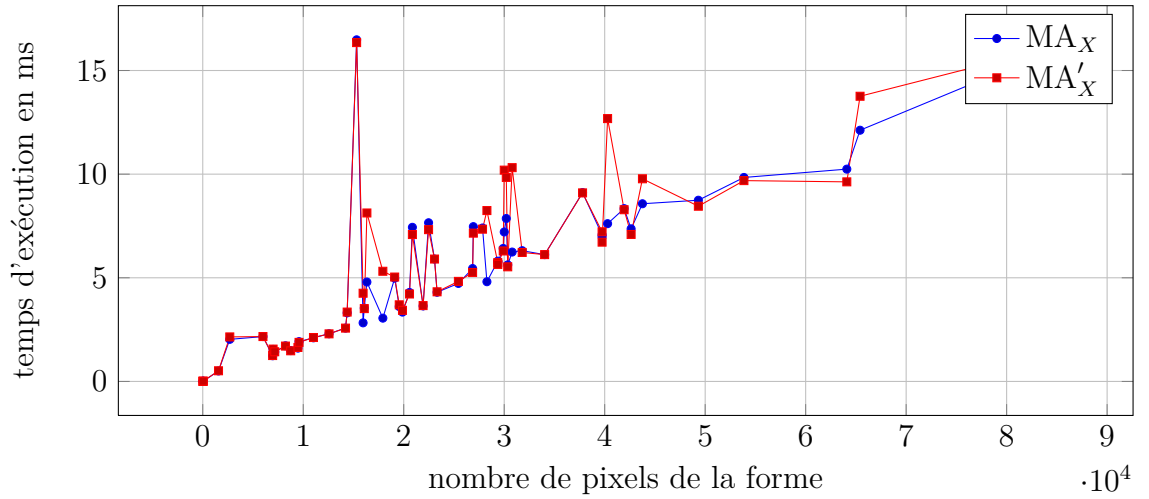


FIGURE 7.32 – Temps de calcul de MA_X et de MA'_X en d_8 en fonction du nombre de pixels de la forme.

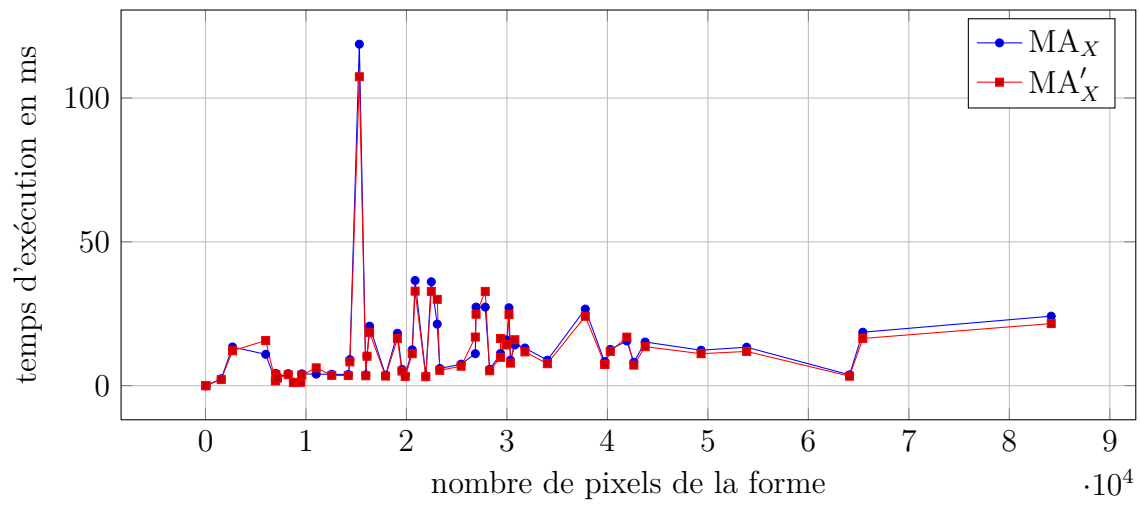


FIGURE 7.33 – Temps de calcul de MA_X et de MA'_X en d_{oct} en fonction du nombre de pixels de la forme.

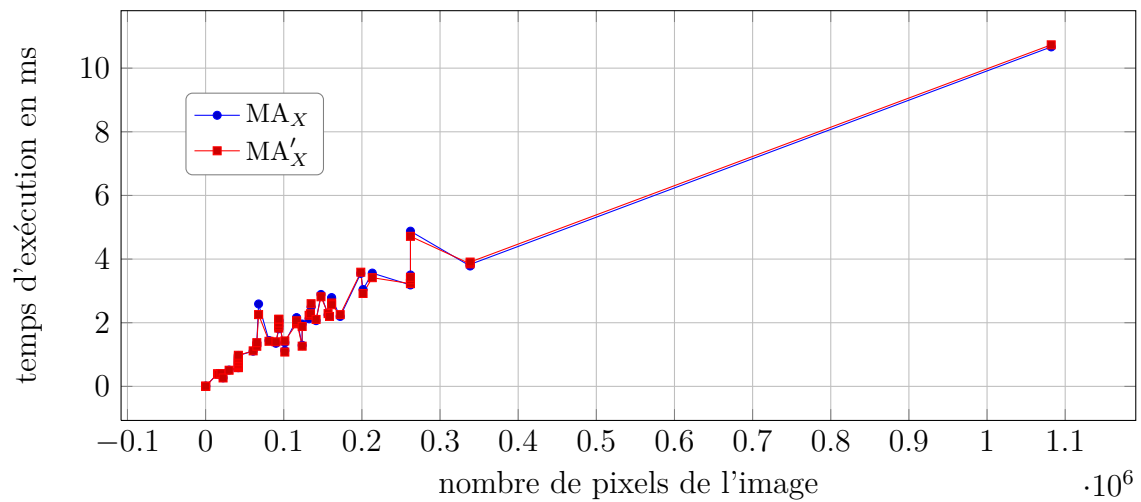


FIGURE 7.34 – Temps de calcul de MA_X et de MA'_X en d_4 en fonction du nombre de pixels de l'image.

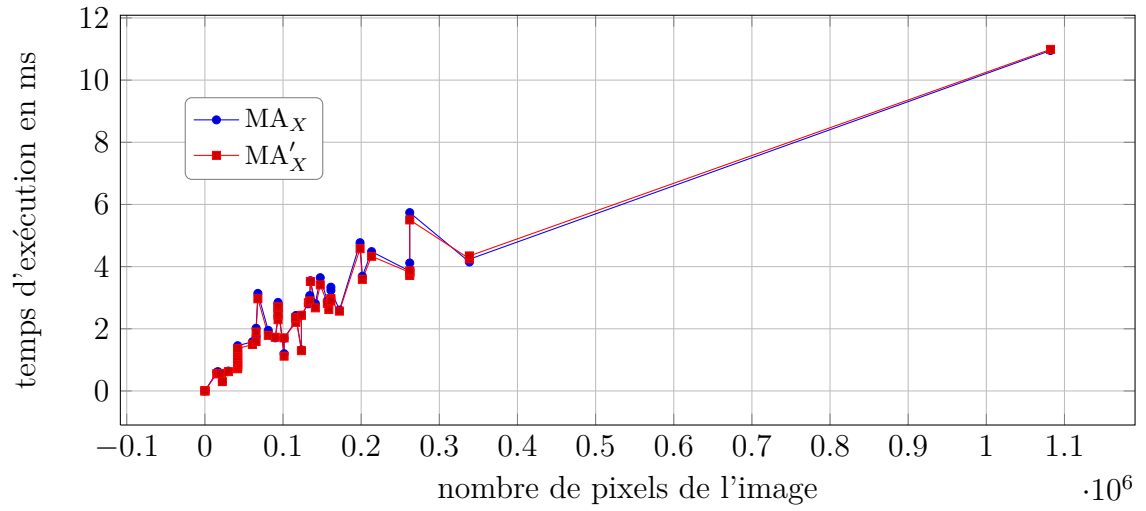


FIGURE 7.35 – Temps de calcul de MA_X et de MA'_X en d_8 en fonction du nombre de pixels de l'image.

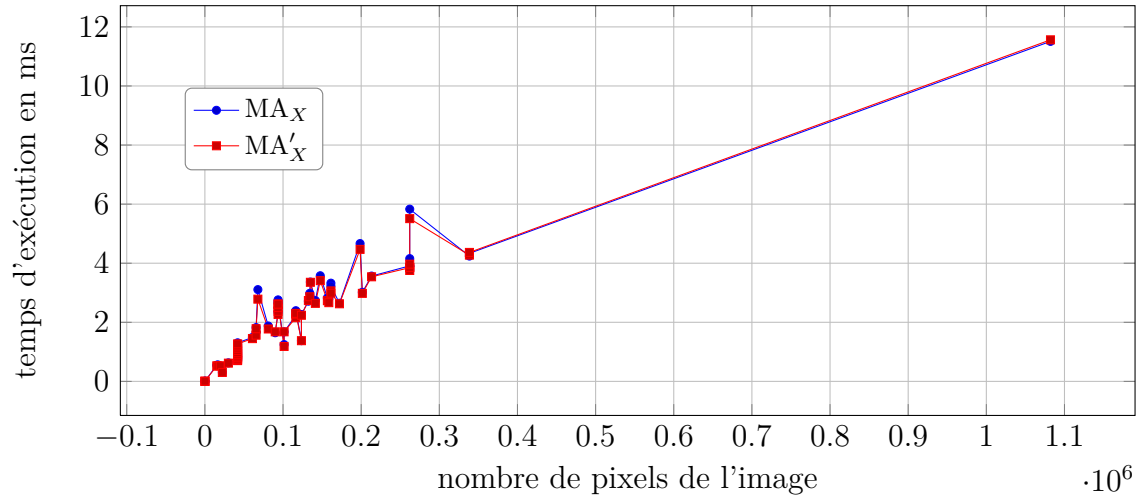


FIGURE 7.36 – Temps de calcul de MA_X et de MA'_X en d_{oct} en fonction du nombre de pixels de l'image.

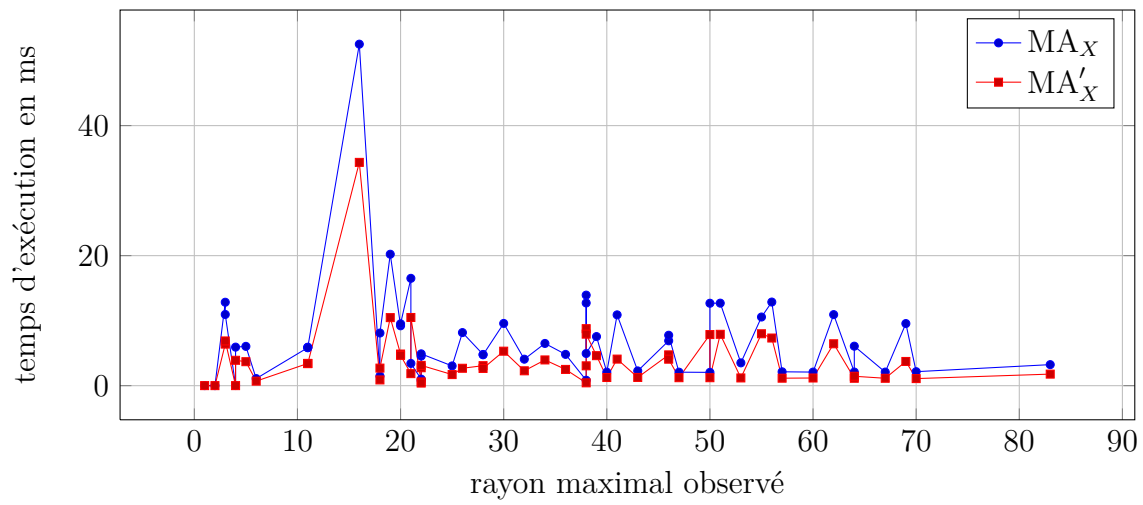


FIGURE 7.37 – Temps de calcul de MA_X et de MA'_X en d_4 en fonction du rayon maximal observé dans l'image.

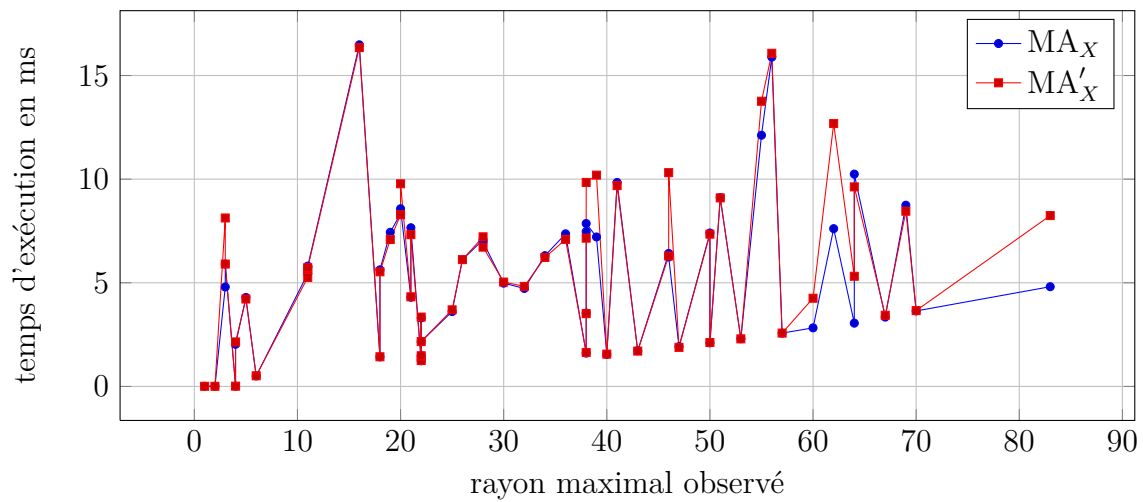


FIGURE 7.38 – Temps de calcul de MA_X et de MA'_X en d_8 en fonction du rayon maximal observé dans l'image.

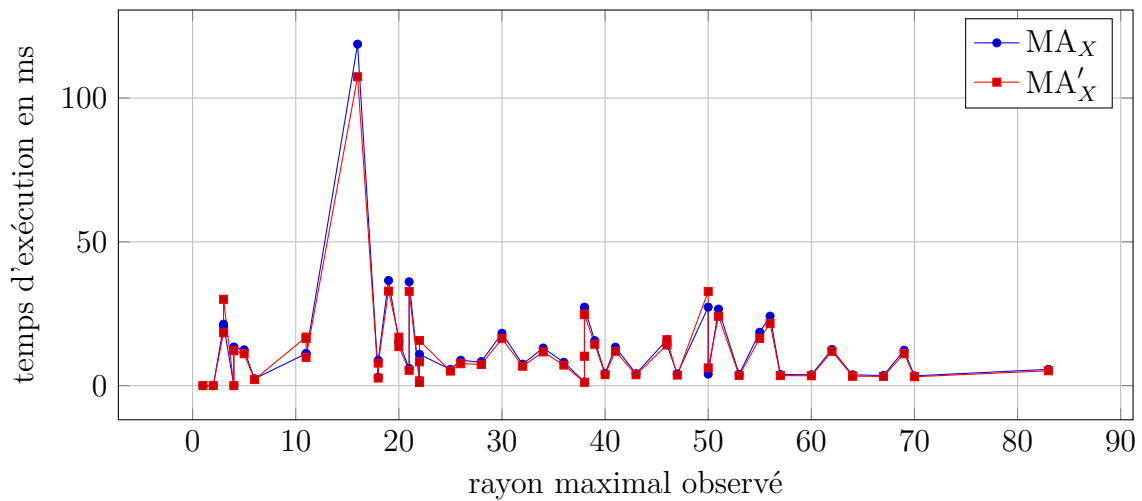


FIGURE 7.39 – Temps de calcul de MA_X et de MA'_X en d_{oct} en fonction du rayon maximal observé dans l'image.

1.4 Squelettisation

Plusieurs courbes de temps de calcul du squelette sont présentées ici. Dans les sous-sections 5.4.1 et 5.4.2 nous avons mis en œuvre sous forme d'option des traitements qui permettent d'affiner le squelette. Nous avons effectué un dernier amincissement du squelette calculé (option -a). Nous avons aussi développé une option qui ré-échantillonne l'image d'entrée (option -e). Cette dernière ne permet pas d'obtenir un squelette fin quand le squelette est orienté en diagonal. C'est pourquoi nous avons aussi relevé les temps de calcul du squelette sur une image ré-échantillonnée puis amincie (option -ae).

Dans un premier temps ces temps de calculs sont présentés en fonction du nombre de pixels de la forme dans l'image. Le tableau 1.4 récapitule les graphiques présentés en fonction de leur distance et de l'option utilisée.

	Sans option	option -a	option -e	option -ae
d_4	7.40	7.43	7.46	7.49
d_8	7.41	7.44	7.47	7.50
d_{oct}	7.42	7.45	7.48	7.51

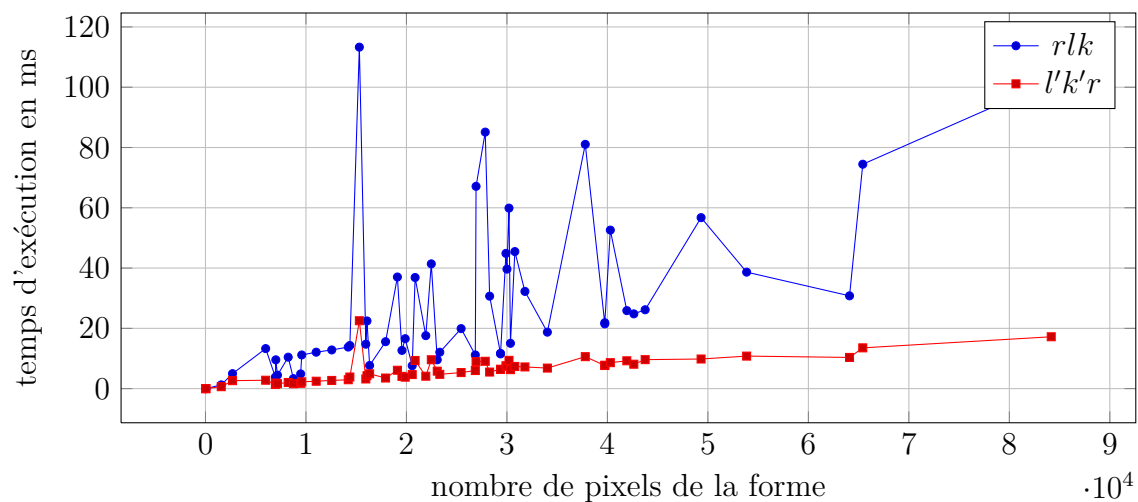


FIGURE 7.40 – Temps de calcul des algorithmes de squelettisation en d_4 en fonction du nombre de pixels de la forme.

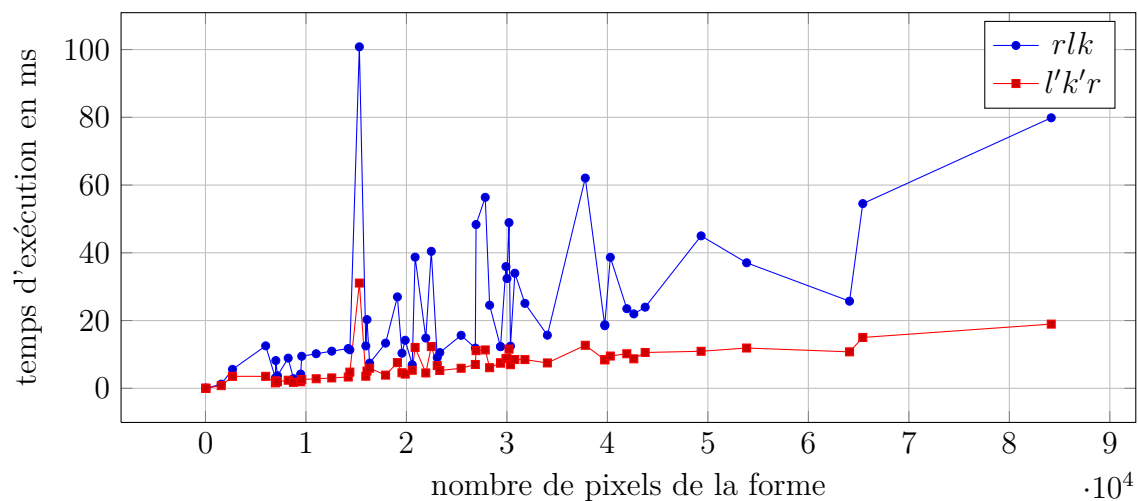


FIGURE 7.41 – Temps de calcul des algorithmes de squelettisation en d_8 en fonction du nombre de pixels de la forme.

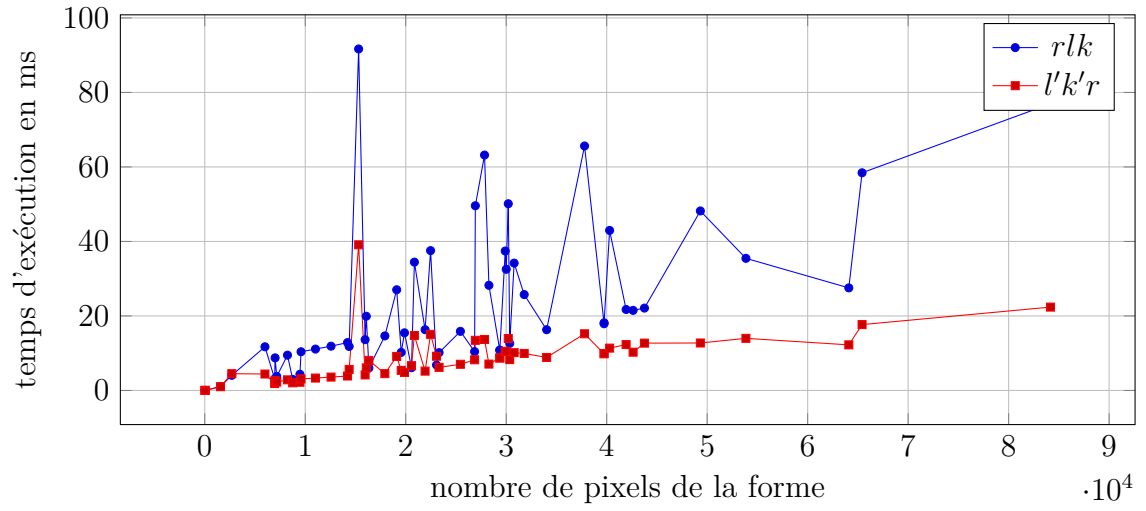


FIGURE 7.42 – Temps de calcul des algorithmes de squelettisation en d_{oct} en fonction du nombre de pixels de la forme.

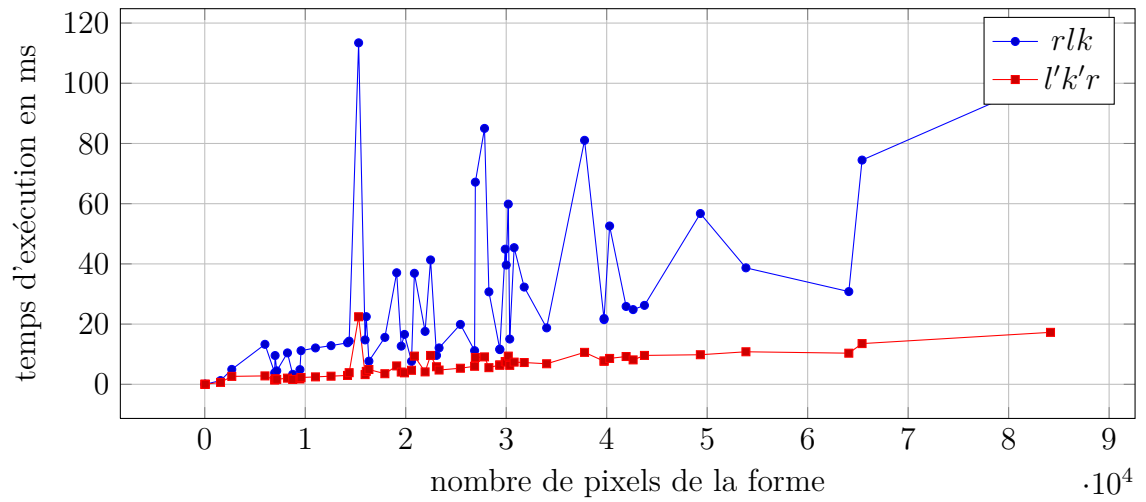


FIGURE 7.43 – Temps de calcul des algorithmes de squelettisation en d_4 après amincissement en fonction du nombre de pixels de la forme.

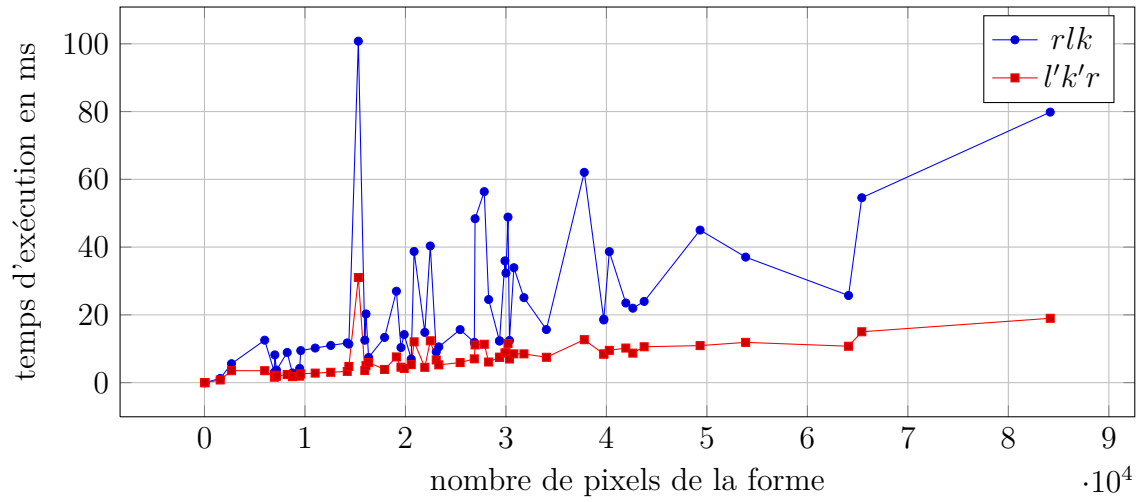


FIGURE 7.44 – Temps de calcul des algorithmes de squelettisation en d_8 après amincissement en fonction du nombre de pixels de la forme.

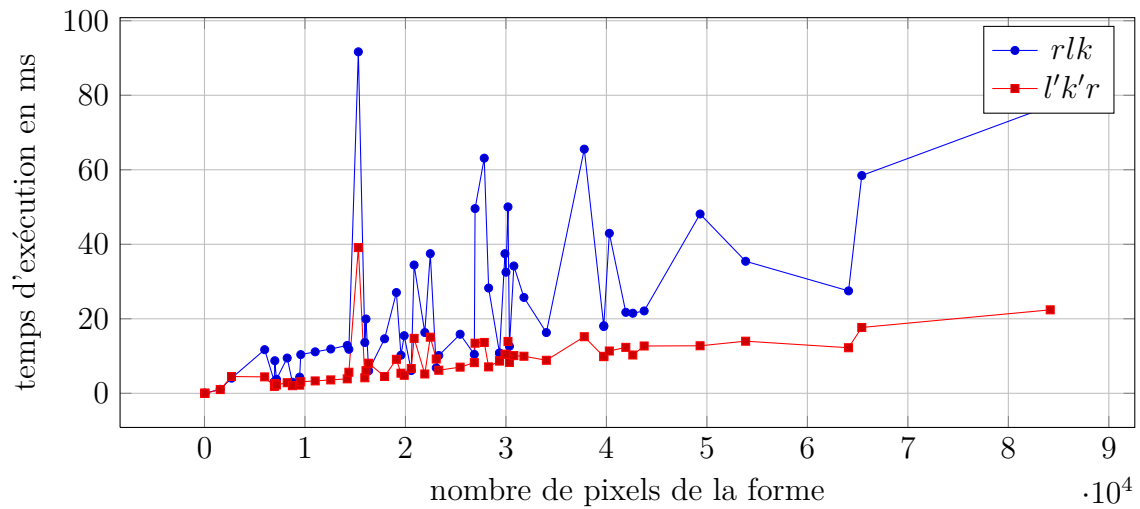


FIGURE 7.45 – Temps de calcul des algorithmes de squelettisation en d_{oct} après amincissement en fonction du nombre de pixels de la forme.

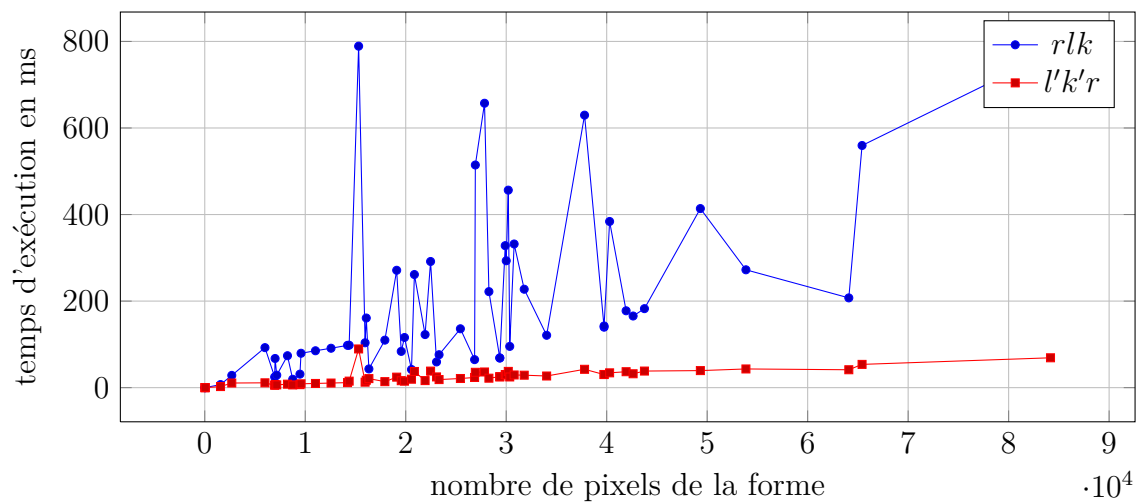


FIGURE 7.46 – Temps de calcul des algorithmes de squelettisation en d_4 après agrandissement de l'image en fonction du nombre de pixels de la forme.

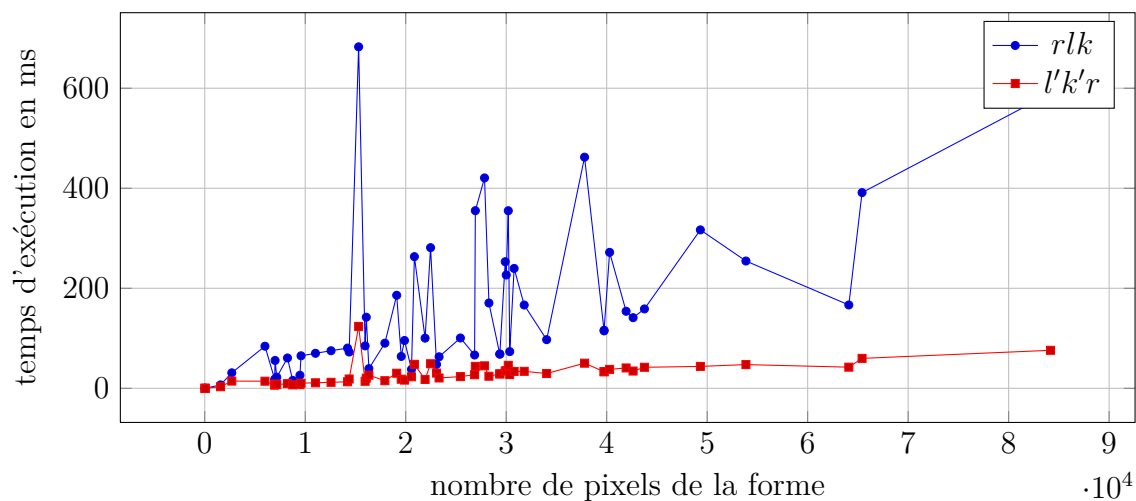


FIGURE 7.47 – Temps de calcul des algorithmes de squelettisation en d_8 après agrandissement de l'image en fonction du nombre de pixels de la forme.

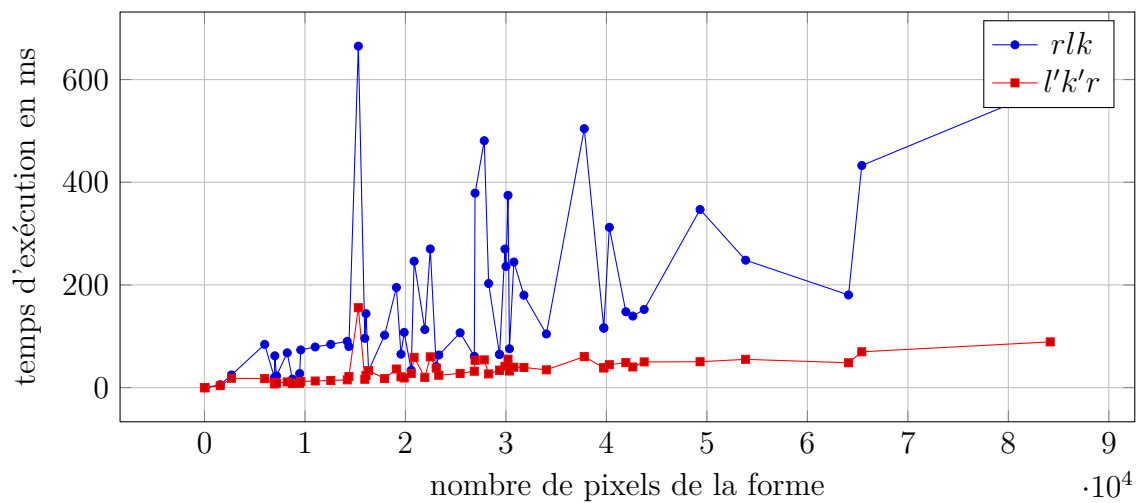


FIGURE 7.48 – Temps de calcul des algorithmes de squelettisation en d_{oct} après agrandissement de l'image en fonction du nombre de pixels de la forme.

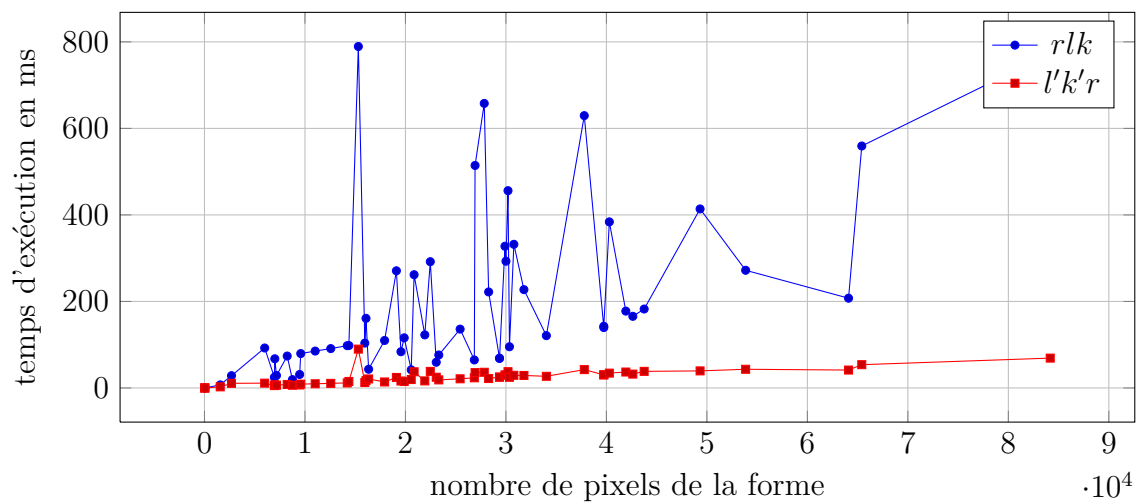


FIGURE 7.49 – Temps de calcul des algorithmes de squelettisation en d_4 après amincissement et agrandissement de l'image en fonction du nombre de pixels de la forme.

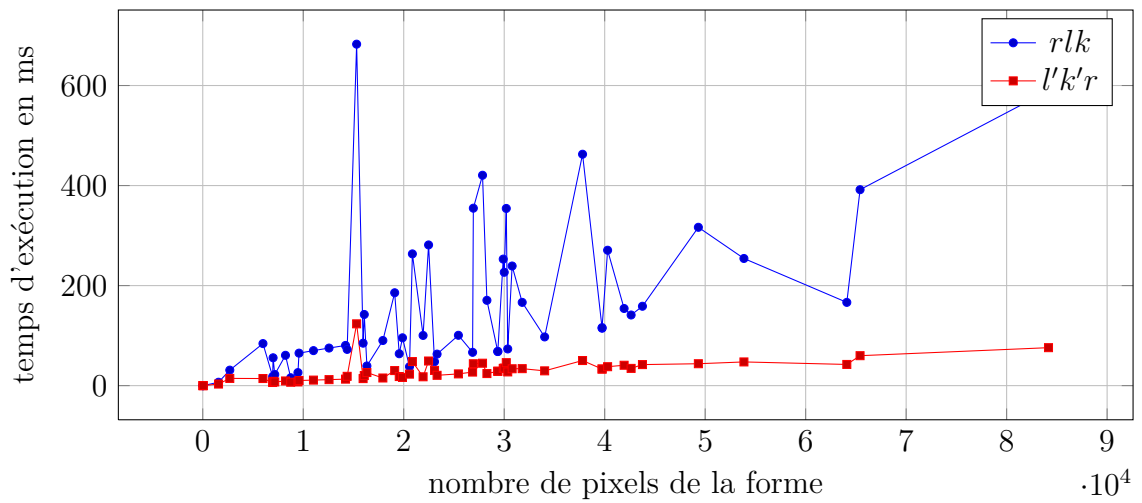


FIGURE 7.50 – Temps de calcul des algorithmes de squelettisation en d_8 après amincissement et agrandissement de l'image en fonction du nombre de pixels de la forme.

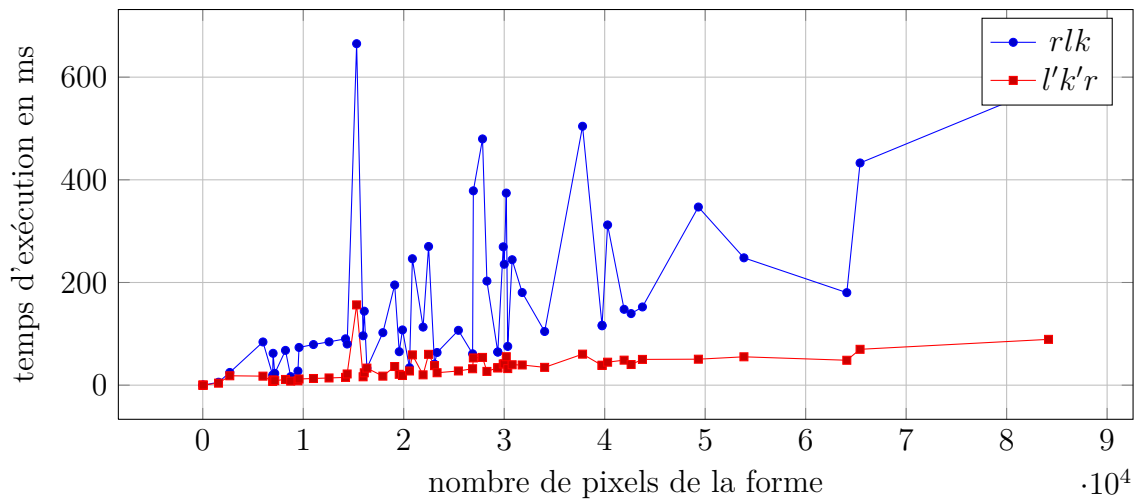
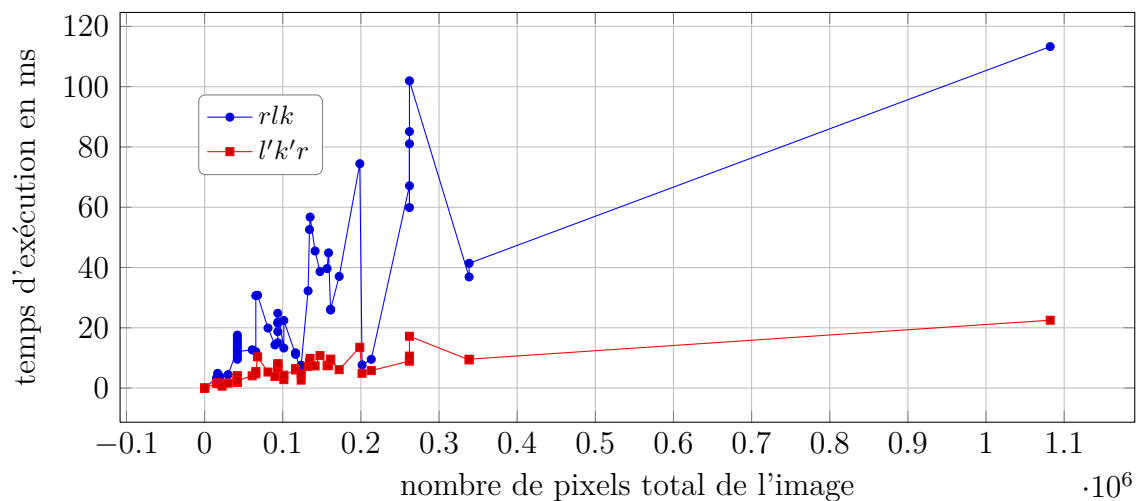
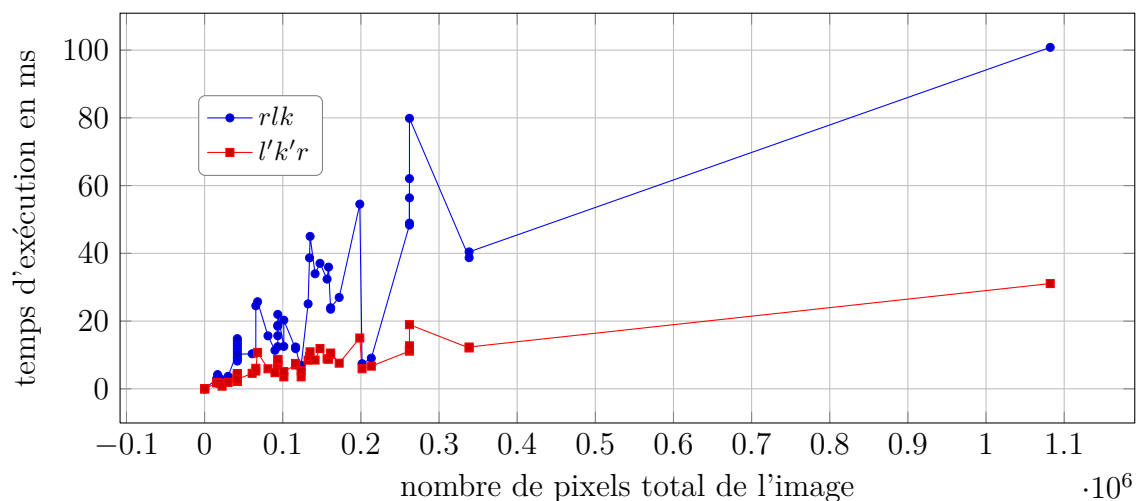


FIGURE 7.51 – Temps de calcul des algorithmes de squelettisation en d_{oct} après amincissement et agrandissement de l'image en fonction du nombre de pixels de la forme.

Dans un second temps, les temps d'exécution du calcul du squelette sont présentés en fonction du nombre total de pixels dans l'image (fond + forme). Le tableau 1.4 récapitule les graphiques présentés en fonction de leur distance et de l'option utilisée.

	Sans option	option -a	option -e	option -ae
d_4	7.52	7.55	7.58	7.61
d_8	7.53	7.56	7.59	7.62
d_{oct}	7.54	7.57	7.60	7.63


FIGURE 7.52 – Temps de calcul des algorithmes de squelettisation en d_4 .

FIGURE 7.53 – Temps de calcul des algorithmes de squelettisation en d_8 .

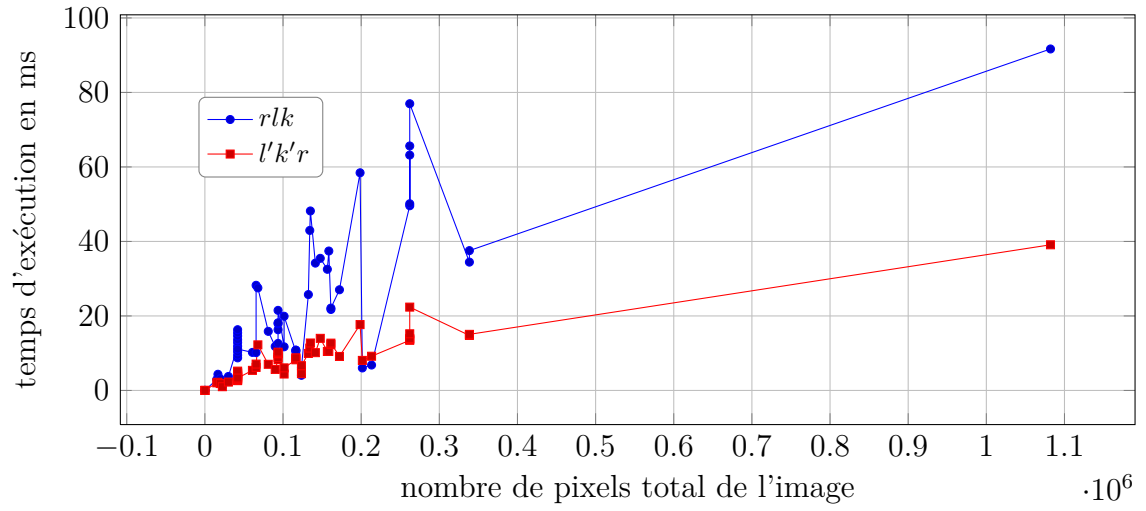


FIGURE 7.54 – Temps de calcul des algorithmes de squelettisation en d_{oct} .

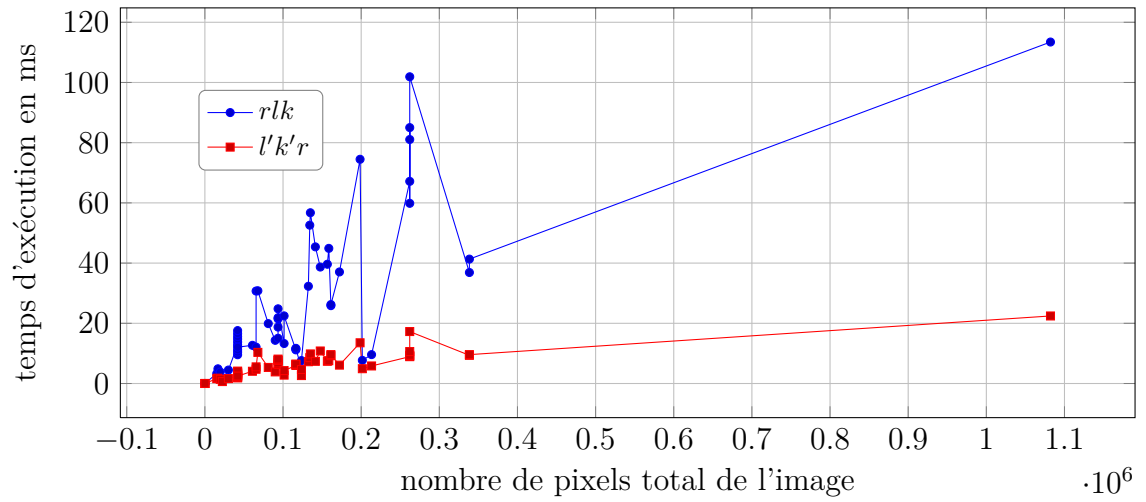


FIGURE 7.55 – Temps de calcul des algorithmes de squelettisation en d_4 avec un dernier amincissement.

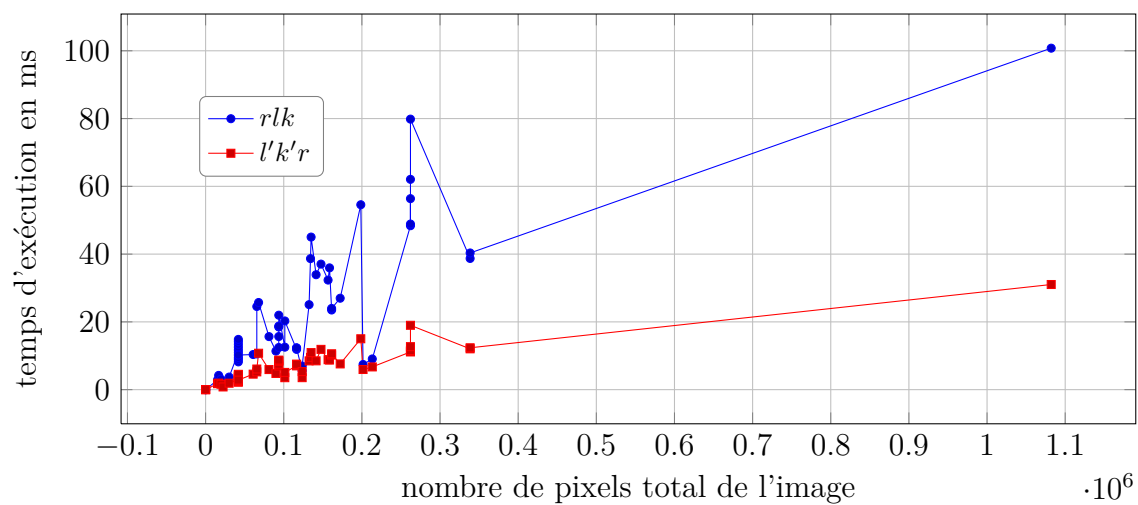


FIGURE 7.56 – Temps de calcul des algorithmes de squelettisation en d_8 avec un dernier amincissement.

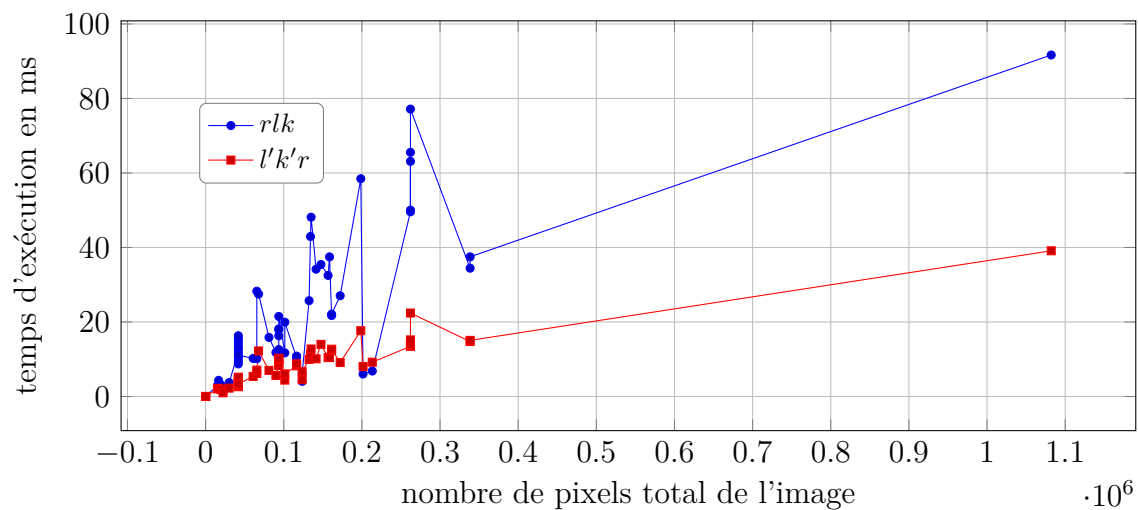


FIGURE 7.57 – Temps de calcul des algorithmes de squelettisation en d_{oct} avec un dernier amincissement.

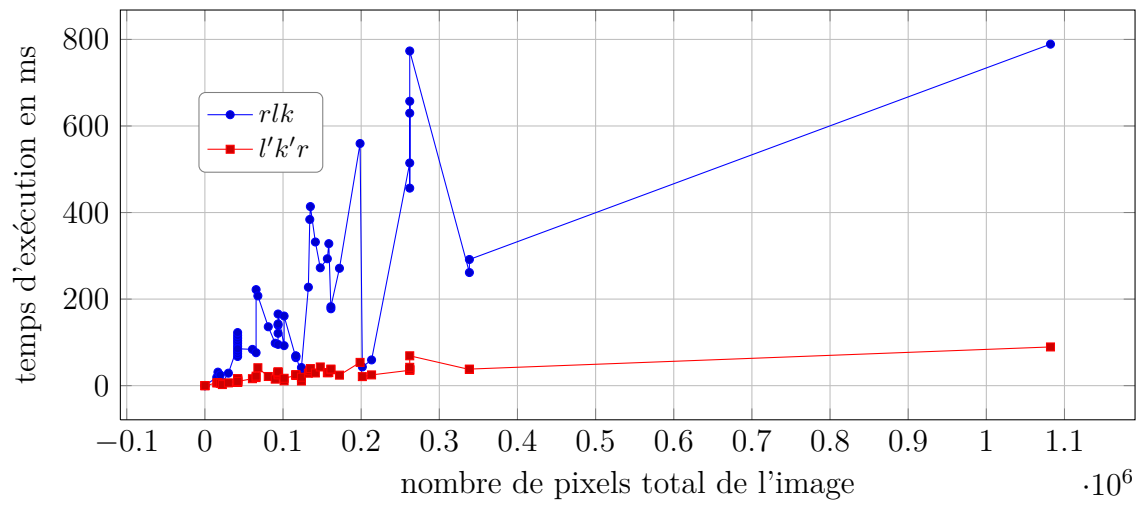


FIGURE 7.58 – Temps de calcul des algorithmes de squelettisation en d_4 sur des images sur-échantillonnées.

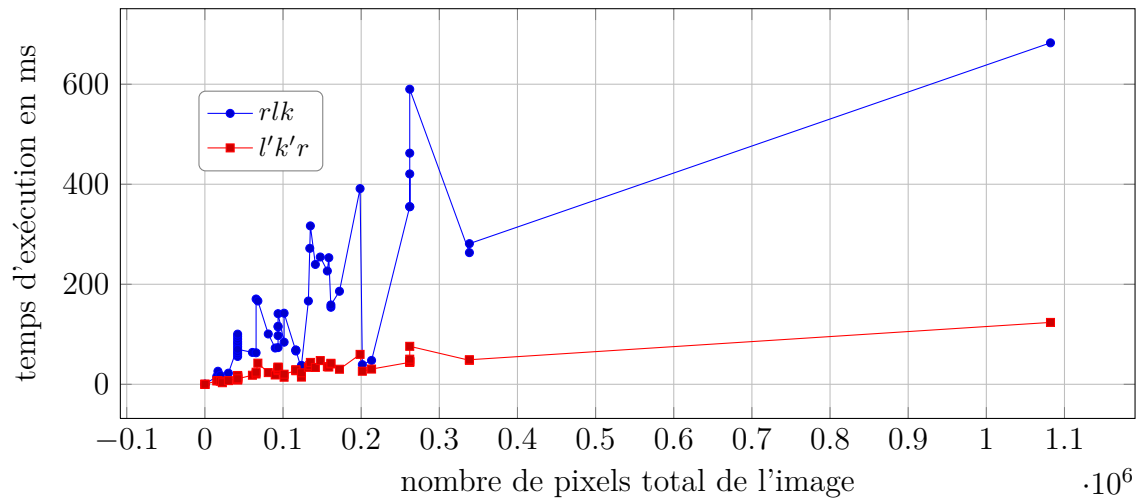


FIGURE 7.59 – Temps de calcul des algorithmes de squelettisation en d_8 sur des images sur-échantillonnées.

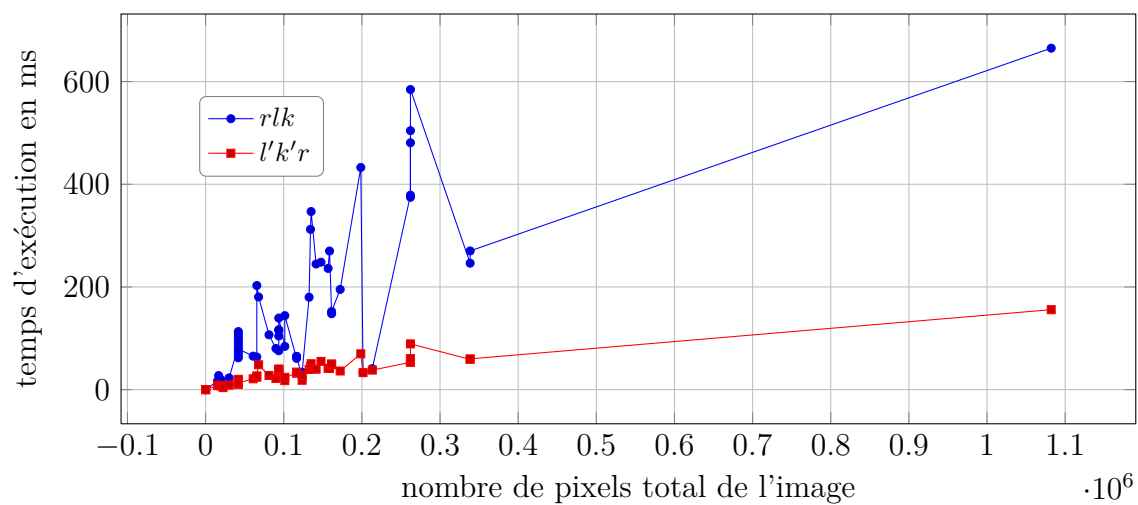


FIGURE 7.60 – Temps de calcul des algorithmes de squelettisation en d_{oct} sur des images sur-échantillonnées.

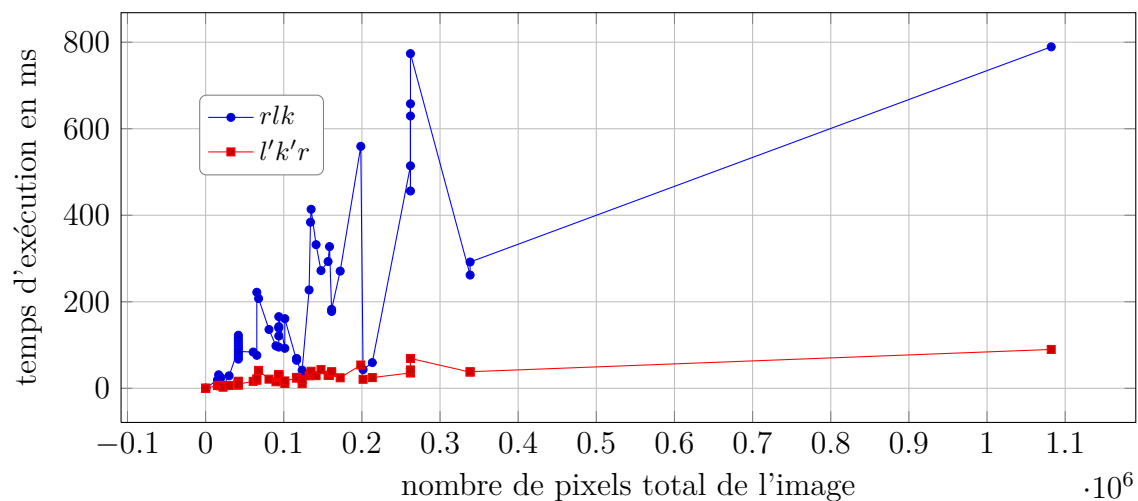


FIGURE 7.61 – Temps de calcul des algorithmes de squelettisation en d_4 sur des images sur-échantillonnées avec un dernier amincissement.

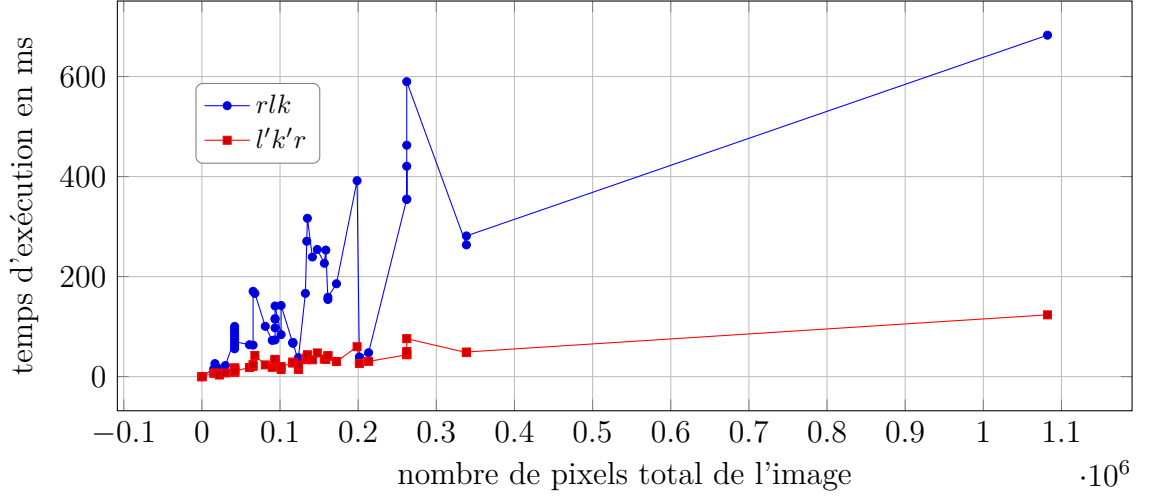


FIGURE 7.62 – Temps de calcul des algorithmes de squelettisation en d_8 sur des images sur-échantillonnées avec un dernier amincissement.

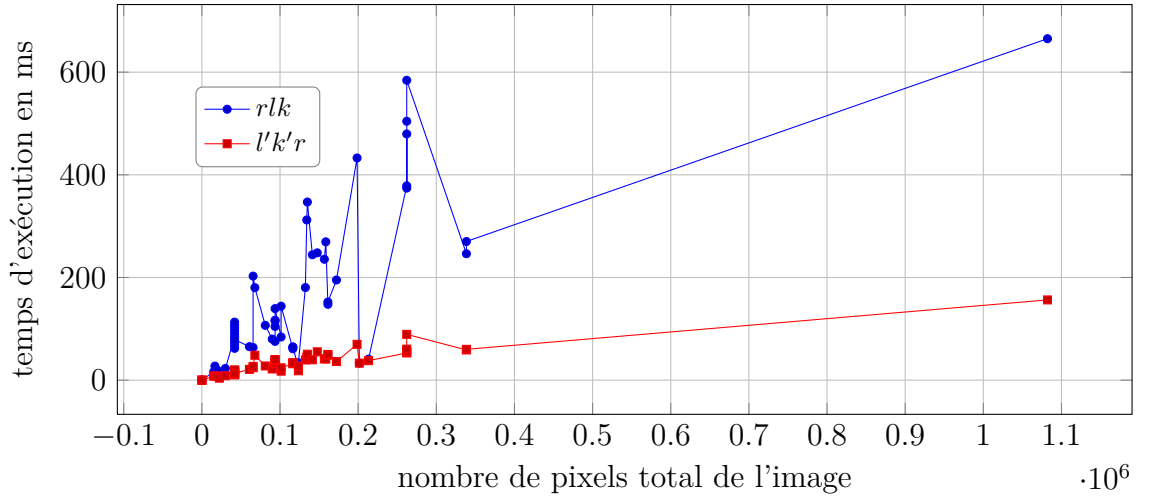


FIGURE 7.63 – Temps de calcul des algorithmes de squelettisation en d_{oct} sur des images sur-échantillonnées avec un dernier amincissement.

2 Extraction des caractéristiques osseuses

Nous avons présenté quelques caractéristiques extraites par les praticiens du LIOAD. Dans un premier temps, la section 2.1 présente une image extraite du

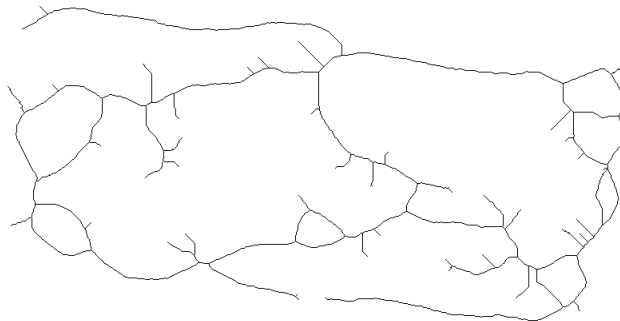
volume d'images, son squelette et les informations extraites par notre algorithme. Puis dans un second temps, la section 2.2 présente toutes les caractéristiques extraites, par le logiciel utilisé au LIOAD, à partir du volume d'images.

2.1 Caractéristiques extraites d'une image d'os.

L'os étudié dans cette section est présenté dans la figure 7.64 avec son squelette.



(a) Os trabéculaire



(b) Squelette de l'os

FIGURE 7.64 – Os trabéculaire binarisé et son squelette.

Le tableau 7.6 présente les caractéristiques extraites de l'os présenté précédemment. On note un taux d'erreur de 18,7% dans la détection des points spéciaux. Effectivement la somme, par trabécules, du nombre de points d'intersection et terminaux doit être de 2 ce qui n'a pas toujours observé.

ID	Largeur	Écart type Largeur	Longueur	nombre de trabécules voisines	nombre de points d'intersection	nombre de points terminaux
1	7,17	0,69	6	2	1	1
2	8,00	0,00	1	2	1	1
3	8,00	0,00	10	4	2	0
4	9,25	0,43	4	4	2	0
5	9,00	0,00	1	2	1	1
6	7,00	0,00	1	2	1	1
7	8,00	0,00	10	4	2	0
8	10,25	0,43	4	4	2	0
9	7,00	0,00	1	2	1	1
10	7,00	0,00	2	4	2	0
11	9,33	0,47	6	0	0	0
12	9,00	0,00	1	2	1	1
13	6,00	0,00	1	2	1	1
14	7,00	0,00	4	4	2	0
15	9,83	0,37	6	4	2	0
16	6,00	0,00	1	2	1	1
17	6,00	0,00	4	4	2	0
18	9,25	0,43	4	0	0	0
19	9,00	0,00	1	2	1	1
20	6,00	0,00	3	4	2	0
21	10,50	0,50	2	5	2	0
22	10,00	0,00	11	0	0	0
23	5,50	0,50	8	0	0	0
24	10,00	0,00	1	3	1	1
25	10,00	0,00	6	5	2	0
26	9,00	0,00	1	2	1	1
27	5,00	0,00	1	2	1	1
28	11,67	0,47	3	3	1	1
29	9,17	0,37	6	4	2	0
30	9,00	0,00	1	2	1	1
31	5,00	0,00	5	4	2	0
32	8,63	0,48	8	4	2	0
33	8,00	0,00	1	2	1	1
34	7,67	0,47	6	4	2	0
35	7,00	0,00	1	2	1	1
36	7,50	0,50	6	4	2	0
37	7,00	0,00	1	2	1	1
38	6,30	0,46	10	4	2	0
39	5,00	0,00	1	2	1	1

2. *Extraction des caractéristiques osseuses*

ID	Largeur	Écart type Largeur	Longueur	nombre de trabécules voisines	nombre de points d'intersection	nombre de points terminaux
40	4,00	0,00	1	2	1	1
41	5,70	0,46	10	4	2	0
42	5,00	0,00	1	2	1	1
43	3,00	0,00	3	2	1	1
44	5,00	0,00	2	5	2	0
45	6,00	0,00	3	4	2	0
46	6,00	0,00	1	2	1	1
47	3,44	0,50	9	4	2	1
48	3,00	0,00	1	2	1	1
49	2,23	0,58	13	4	2	1
50	2,00	0,00	1	2	1	1
51	4,00	0,00	1	3	1	1
52	4,00	0,00	3	0	0	0
53	3,50	0,50	4	4	2	1
54	4,00	0,00	1	0	0	0
55	3,00	0,00	1	0	0	0
56	1,17	0,37	18	4	2	0
57	4,00	0,00	1	3	1	1
58	6,67	0,47	9	4	2	0
59	4,00	0,00	1	2	1	1
60	2,00	0,00	1	2	1	1
61	2,00	0,00	1	2	1	1
62	3,00	0,00	1	2	1	1
63	3,00	0,00	1	2	1	1
64	4,14	0,35	7	4	2	0
65	4,00	0,00	1	2	1	1
66	5,00	0,00	7	4	2	0
67	5,00	0,00	1	2	1	1
68	15,78	4,94	36	2	1	1
69	4,00	0,00	3	5	2	0
70	4,27	0,45	11	4	2	1
71	4,00	0,00	1	0	0	0
72	2,00	0,00	1	4	2	0
73	3,00	0,00	1	4	2	0
74	3,00	0,00	1	4	2	0
75	4,00	0,00	1	0	0	0
76	7,00	0,00	16	0	0	0
77	4,00	0,00	1	2	1	1
78	5,00	0,00	2	0	0	0

ID	Largeur	Écart type Largeur	Longueur	nombre de trabécules voisines	nombre de points d'intersection	nombre de points terminaux
79	14, 11	5, 06	36	4	2	0
80	3, 00	0, 00	1	2	1	1
81	3, 00	0, 00	3	0	0	0
82	4, 83	0, 69	6	4	2	2
83	2, 00	0, 00	1	2	1	1
84	3, 00	0, 00	2	4	2	0
85	6, 00	0, 00	1	2	1	1
86	5, 00	0, 00	14	0	0	0
87	2, 00	0, 00	1	2	1	1
88	2, 00	0, 00	2	4	2	0
89	6, 33	0, 47	6	4	2	0
90	1, 00	0, 00	2	2	1	1
91	1, 00	0, 00	1	2	1	1
92	2, 00	0, 00	2	4	2	0
93	7, 00	0, 00	4	0	0	0
94	6, 00	0, 00	1	2	1	1
95	5, 14	0, 43	29	4	2	1
96	5, 00	0, 00	4	0	0	0
97	6, 00	0, 00	1	4	2	0
98	6, 00	0, 00	5	0	0	0
99	5, 00	0, 00	5	0	0	0
100	5, 00	0, 00	13	0	0	0
101	7, 50	0, 50	4	2	1	1
102	6, 30	1, 90	10	2	1	1
103	24, 00	0, 00	1	2	1	1
104	25, 00	0, 00	6	4	2	0
105	25, 15	0, 43	73	4	2	0
106	24, 00	0, 00	6	4	2	0
107	6, 59	2, 14	17	2	1	1
108	20, 50	1, 71	12	4	2	0
109	26, 14	0, 35	7	0	0	0
110	27, 00	0, 00	4	0	0	0
111	27, 00	0, 00	1	2	1	1
112	28, 50	0, 50	4	4	2	0
113	28, 00	0, 00	7	0	0	0
114	28, 20	0, 40	5	0	0	0
115	29, 00	0, 00	4	0	0	0
116	29, 00	0, 00	1	2	1	1
117	30, 14	0, 99	7	4	2	0

2. *Extraction des caractéristiques osseuses*

ID	Largeur	Écart type Largeur	Longueur	nombre de trabécules voisines	nombre de points d'intersection	nombre de points terminaux
118	29,64	0,48	11	0	0	0
119	29,71	0,70	7	0	0	0
120	6,35	3,53	37	2	1	1
121	18,00	0,00	1	2	1	1
122	17,00	0,00	1	2	1	1
123	18,00	0,00	2	4	2	0
124	31,00	0,00	5	0	0	0
125	17,00	0,00	1	2	1	1
126	17,00	0,00	2	4	2	0
127	31,06	0,54	17	0	0	0
128	17,43	0,49	7	5	2	1
129	17,00	0,00	1	0	0	0
130	17,00	0,00	1	3	1	1
131	18,00	0,00	2	0	0	0
132	32,00	0,00	1	2	1	1
133	17,00	0,00	1	3	1	1
134	32,84	0,59	19	4	2	0
135	6,57	3,41	23	2	1	1
136	10,00	0,00	1	2	1	1
137	11,00	0,00	6	4	2	0
138	11,12	0,32	17	4	2	0
139	17,25	0,43	4	5	2	1
140	25,00	0,00	1	2	1	1
141	24,57	0,49	7	4	2	0
142	24,00	0,00	1	2	1	1
143	10,00	0,00	1	2	1	1
144	10,00	0,00	4	4	2	0
145	11,00	0,00	1	2	1	1
146	17,00	0,00	1	0	0	0
147	24,25	0,43	4	4	2	0
148	24,00	0,00	1	2	1	1
149	8,81	2,44	21	2	1	1
150	9,00	0,00	1	2	1	1
151	10,00	0,00	2	4	2	0
152	12,00	0,00	1	2	1	1
153	16,00	0,00	1	2	1	1
154	28,50	1,86	10	4	2	0
155	25,00	0,00	2	4	2	0
156	24,00	0,00	1	2	1	1

ID	Largeur	Écart type Largeur	Longueur	nombre de trabécules voisines	nombre de points d'intersection	nombre de points terminaux
157	9,00	0,00	3	4	2	0
158	12,00	0,00	2	4	2	0
159	12,00	0,00	1	2	1	1
160	16,00	0,00	1	2	1	1
161	16,00	0,00	1	4	2	0
162	24,25	0,43	4	4	2	0
163	24,00	0,00	1	2	1	1
164	8,00	0,00	1	2	1	1
165	13,00	0,00	2	4	2	0
166	13,00	0,00	1	2	1	1
167	13,00	0,00	1	2	1	1
168	14,14	0,35	7	4	2	0
169	34,00	0,00	3	0	0	0
170	33,63	0,48	8	4	2	0
171	33,00	0,00	1	2	1	1
172	25,00	0,00	2	4	2	0
173	25,00	0,00	1	2	1	1
174	8,00	0,00	1	2	1	1
175	8,00	0,00	2	4	2	0
176	13,00	0,00	2	4	2	0
177	13,50	0,50	2	4	2	0
178	14,00	0,00	1	2	1	1
179	15,00	0,00	1	2	1	1
180	15,33	0,47	6	4	2	0
181	16,00	0,00	1	4	2	0
182	38,50	2,63	18	4	2	0
183	32,90	0,30	20	4	2	0
184	32,00	0,00	1	2	1	1
185	24,50	0,50	4	4	2	0
186	24,00	0,00	1	2	1	1
187	7,00	0,00	1	2	1	1
188	8,00	0,00	2	4	2	0
189	15,00	0,00	1	4	2	0
190	33,00	0,00	10	4	2	0
191	32,00	0,00	1	2	1	1
192	25,00	0,00	2	4	2	0
193	25,00	0,00	1	2	1	1
194	7,00	0,00	1	2	1	1
195	7,00	0,00	2	4	2	0

2. *Extraction des caractéristiques osseuses*

ID	Largeur	Écart type Largeur	Longueur	nombre de trabécules voisines	nombre de points d'intersection	nombre de points terminaux
196	20,50	2,27	26	4	2	0
197	31,80	0,40	10	4	2	0
198	31,00	0,00	1	2	1	1
199	24,33	0,47	6	4	2	0
200	24,00	0,00	1	2	1	1
201	4,33	1,15	9	2	1	1
202	6,00	0,00	1	3	1	1
203	7,00	0,00	4	5	2	0
204	32,00	0,00	10	4	2	0
205	32,00	0,00	1	2	1	1
206	24,00	0,00	2	4	2	0
207	24,00	0,00	1	2	1	1
208	6,00	0,00	1	3	1	1
209	31,40	0,49	10	4	2	0
210	31,00	0,00	1	2	1	1
211	24,50	0,50	2	4	2	0
212	24,00	0,00	1	2	1	1
213	6,33	0,47	3	5	2	0
214	32,00	0,00	8	4	2	0
215	32,00	0,00	1	2	1	1
216	24,00	0,00	2	4	2	0
217	24,00	0,00	1	2	1	1
218	31,67	0,47	6	4	2	0
219	24,00	0,00	2	4	2	0
220	23,00	0,00	1	2	1	1
221	5,00	0,00	1	2	1	1
222	6,00	0,00	3	0	0	0
223	23,00	0,00	2	4	2	0
224	23,00	0,00	1	2	1	1
225	9,17	0,69	6	2	1	1
226	5,00	0,00	1	2	1	1
227	5,00	0,00	4	4	2	0
228	34,58	1,38	12	4	2	0
229	23,00	0,00	2	4	2	0
230	22,00	0,00	1	2	1	1
231	5,50	0,71	8	4	2	3
232	22,00	0,00	2	4	2	0
233	22,00	0,00	1	2	1	1
234	5,14	0,35	7	0	0	0

ID	Largeur	Écart type Largeur	Longueur	nombre de trabécules voisines	nombre de points d'intersection	nombre de points terminaux
235	22,00	0,00	3	4	2	0
236	6,00	0,00	5	0	0	0
237	21,00	0,00	1	2	1	1
238	6,57	0,73	7	4	2	2
239	6,00	0,00	3	0	0	0
240	8,00	0,00	1	2	1	1
241	8,00	0,00	1	2	1	1
242	7,00	0,00	7	0	0	0
243	25,07	5,11	27	4	2	0
244	9,30	1,10	10	4	2	2
245	8,00	0,00	2	4	2	0
246	7,76	0,42	17	0	0	0
247	11,25	0,43	4	4	2	1
248	11,25	0,43	4	4	2	1
249	11,88	0,33	8	0	0	0
250	12,00	0,00	1	2	1	1
251	10,20	0,40	5	0	0	0
252	11,00	0,00	1	2	1	1
253	11,43	0,49	7	0	0	0
254	12,00	0,00	1	2	1	1
255	12,00	0,00	1	0	0	0
256	12,00	0,00	1	2	1	1
257	12,00	0,00	2	4	2	0
258	13,83	0,69	12	4	2	0
259	12,00	0,00	1	2	1	1
260	12,00	0,00	2	4	2	0
261	5,96	2,73	25	2	1	1
262	11,00	0,00	1	2	1	1
263	12,00	0,00	2	4	2	0
264	15,00	0,00	7	0	0	0
265	14,50	0,50	4	4	2	0
266	13,00	0,00	2	4	2	0
267	34,18	15,50	44	2	1	1
268	49,79	3,54	28	4	2	0
269	11,00	0,00	1	2	1	1
270	11,00	0,00	2	4	2	0
271	13,00	0,00	1	2	1	1
272	13,50	0,50	2	4	2	0
273	37,00	0,00	1	2	1	1

ID	Largeur	Écart type Largeur	Longueur	nombre de trabécules voisines	nombre de points d'intersection	nombre de points terminaux
274	10,00	0,00	1	2	1	1
275	11,00	0,00	2	4	2	0
276	14,33	0,47	9	0	0	0
277	14,00	0,00	1	0	0	0
278	38,33	0,47	3	4	2	0
279	10,00	0,00	1	2	1	1
280	10,00	0,00	2	4	2	0
281	14,00	0,00	1	2	1	1
282	14,25	0,43	4	4	2	0
283	14,00	0,00	1	4	2	0
284	9,00	0,00	1	2	1	1
285	10,00	0,00	4	4	2	0
286	15,00	0,00	1	2	1	1
287	9,00	0,00	1	2	1	1
288	9,00	0,00	2	4	2	0
289	15,00	0,00	2	4	2	0
290	14,00	0,00	1	2	1	1
291	39,00	0,00	1	2	1	1
292	8,00	0,00	1	2	1	1
293	9,00	0,00	2	4	2	0
294	16,00	0,00	3	4	2	0
295	16,00	0,00	1	2	1	1
296	15,00	0,00	1	2	1	1
297	15,00	0,00	1	0	0	0
298	40,00	0,00	2	4	2	0
299	8,00	0,00	1	2	1	1
300	8,00	0,00	2	4	2	0

TABLE 7.2 – Largeur et longueur observées sur l'os 7.17.

2.2 Caractéristiques extraites par le logiciel d'odonto

Description	Abréviation	Résultat	Unité
Tissue volume	TV	57,22555	mm^3
Bone volume	BV	39,53954	mm^3
Percent bone volume	BV/TV	69,0942	
Tissue surface	TS	127,4932	mm^2
Bone surface	BS	273,13702	mm^2
Bone surface / volume ratio	BS/BV	6,90795	$1/mm$
Mean total crossectional tissue area	T.Ar	21,75615	mm^2
Mean total crossectional tissue perimeter	T.Pm	20,16092	mm
Mean total crossectional bone area	B.Ar	15,03224	mm^2
Mean total crossectional bone perimeter	B.Pm	58,38801	mm
Mean number of objects per slice	Obj.N	2,32847	
Average object area per slice	Av.Obj.Ar	9,38278	mm^2
Average object equivalent circle diameter per slice	Av.Obj.ECD	3,27301	mm
Mean polar moment of inertia	MMI(polar)	75,03378	mm^4
Mean eccentricity	Ecc	0,83578	
Crossectional thickness	Cs.Th	0,51491	mm
Trabecular thickness (plate model)	Tb.Th(pl)	0,28952	mm
Trabecular separation (plate model)	Tb.Sp(pl)	0,1295	mm
Trabecular number (plate model)	Tb.N(pl)	2,3865	$1/mm$
Trabecular diameter (rod model)	Tb.Dm(rd)	0,57904	mm
Trabecular separation (rod model)	Tb.Sp(rd)	0,03831	mm
Trabecular number (rod model)	Tb.N(rd)	1,61981	$1/mm$
Mean trabecular pattern factor	Tb.Pf	-1,6867	$1/mm$
Closed porosity (percent)	Po(cl)	22,69478	
Centroid (x)	Crd.X	8,17014	mm
Centroid (y)	Crd.Y	11,1517	mm
Centroid (z)	Crd.Z	10,49072	mm
Mean fractal dimension	FD	1,04936	
Total intersection surface	i.S	48,1655	mm^2

TABLE 7.3 – Largeur et longueur observées sur l'os 7.17.

Description	Abréviation	Résultat	Unité
Number of layers		137	
Lower vertical position		9,06211	mm
Upper vertical position		11,67323	mm
Pixel size		19,19939	um
Lower grey threshold		20	
Upper grey threshold		255	
Tissue volume	TV	57,19979	mm ³
Bone volume	BV	39,43129	mm ³
Percent bone volume	BV/TV	68,93607	%
Tissue surface	TS	114,16948	mm ²
Bone surface	BS	229,24168	mm ²
Intersection surface	i.S	93,15771	mm ²
Bone surface / volume ratio	BS/BV	5,8137	1/mm
Bone surface density	BS/TV	4,00774	1/mm
Trabecular pattern factor	Tb.Pf	-1,2197	1/mm
Centroid (x)	Cr.d.X	8,17915	mm
Centroid (y)	Cr.d.Y	11,16222	mm
Centroid (z)	Cr.d.Z	10,49072	mm
Structure model index	SMI	-0,3307	
Trabecular thickness	Tb.Th	0,65245	mm
Trabecular number	Tb.N	1,05657	1/mm
Trabecular separation	Tb.Sp	0,5035	mm
Degree of anisotropy	DA	2,22566 (0,55070)	
Eigenvalue 1		0,05917	
Eigenvalue 2		0,0955	
Eigenvalue 3		0,13169	
Fractal dimension	FD	2,09313	
Number of objects	Obj.N	2	
Number of closed pores	Po.N(cl)	13	
Volume of closed pores	Po.V(cl)	0,03405	mm ³
Surface of closed pores	Po.S(cl)	1,30134	mm ²
Closed porosity (percent)	Po(cl)	0,08628	%
Volume of open pore space	Po.V(op)	17,73445	mm ³
Open porosity (percent)	Po(op)	31,0044	%
Total volume of pore space	Po.V(tot)	17,7685	mm ³
Total porosity (percent)	Po(tot)	31,06393	%

TABLE 7.4 – Largeur et longueur observées sur l'os 7.17.

mm	mm	mm^3	%
0,019– < 0,058	0,038	0,01836	0,0471
0,058– < 0,096	0,077	0,10435	0,2676
0,096– < 0,134	0,115	0,3576	0,9172
0,134– < 0,173	0,154	0,72992	1,872
0,173– < 0,211	0,192	1,0966	2,8125
0,211– < 0,250	0,23	1,28819	3,3039
0,250– < 0,288	0,269	1,40059	3,5921
0,288– < 0,326	0,307	2,01337	5,1637
0,326– < 0,365	0,346	2,14796	5,5089
0,365– < 0,403	0,384	1,66116	4,2604
0,403– < 0,442	0,422	1,74337	4,4713
0,442– < 0,480	0,461	1,78995	4,5907
0,480– < 0,518	0,499	2,26619	5,8122
0,518– < 0,557	0,538	2,0563	5,2738
0,557– < 0,595	0,576	1,4343	3,6786
0,595– < 0,634	0,614	1,73871	4,4593
0,634– < 0,672	0,653	1,59289	4,0853
0,672– < 0,710	0,691	1,01346	2,5992
0,710– < 0,749	0,73	1,10688	2,8388
0,749– < 0,787	0,768	0,90929	2,3321
0,787– < 0,826	0,806	0,82178	2,1076
0,826– < 0,864	0,845	1,30856	3,3561
0,864– < 0,902	0,883	0,78343	2,0093
0,902– < 0,941	0,922	1,11938	2,8709
0,941– < 0,979	0,96	0,65528	1,6806
0,979– < 1,018	0,998	1,53185	3,9288
1,018– < 1,056	1,037	0,38858	0,9966
1,056– < 1,094	1,075	0,32789	0,841
1,094– < 1,133	1,114	0,488	1,2516
1,133– < 1,171	1,152	1,50961	3,8717
1,171– < 1,210	1,19	1,01073	2,5922
1,210– < 1,248	1,229	0,09826	0,252
1,248– < 1,286	1,267	0,12069	0,3095
1,286– < 1,325	1,306	0,10356	0,2656
1,325– < 1,363	1,344	0,36067	0,925
1,363– < 1,402	1,382	0,44782	1,1485
1,402– < 1,440	1,421	1,44505	3,7062

TABLE 7.5 – Largeur et longueur observées sur l'os 7.17.

mm	mm	mm^3	%
0,019– < 0,058	0,04	0,03	0,18
0,058– < 0,096	0,08	0,11	0,65
0,096– < 0,134	0,12	0,28	1,61
0,134– < 0,173	0,15	0,48	2,77
0,173– < 0,211	0,19	0,65	3,77
0,211– < 0,250	0,23	0,64	3,67
0,250– < 0,288	0,27	0,65	3,72
0,288– < 0,326	0,31	0,78	4,48
0,326– < 0,365	0,35	0,81	4,69
0,365– < 0,403	0,38	0,77	4,44
0,403– < 0,442	0,42	1,17	6,74
0,442– < 0,480	0,46	1,36	7,84
0,480– < 0,518	0,5	1,34	7,72
0,518– < 0,557	0,54	1,01	5,8
0,557– < 0,595	0,58	1,62	9,32
0,595– < 0,634	0,61	1,42	8,2
0,634– < 0,672	0,65	1,29	7,43
0,672– < 0,710	0,69	1,04	5,97
0,710– < 0,749	0,73	0,11	0,65
0,749– < 0,787	0,77	0,16	0,93
0,787– < 0,826	0,81	0,23	1,34
0,826– < 0,864	0,85	0,46	2,65
0,864– < 0,902	0,88	0,18	1,02
0,902– < 0,941	0,92	0,22	1,27
0,941– < 0,979	0,96	0,54	3,14

TABLE 7.6 – Largeur et longueur observées sur l'os 7.17.

Production scientifique

Articles en cours de révision

- [58] Nicolas Normand, Robin Strand, Pierre Evenou, **Aurore Arlicot**, “A minimal delay neighborhood-sequence distance transform”, *Image Processing On Line*, 2012, en cours de révision.
- [59] ———, “Minimal-delay distance transform for neighborhood-sequence distances in 2D and 3D”, *Computer Vision and Image Understanding*, 2012, en cours de révision.

Conférences avec comité de lecture

- [2] Yves Amouriq, Pierre Evenou, **Aurore Arlicot**, Nicolas Normand, Pierre Layrolle, Pierre Weiss, Jeanpierre Guédon, “Evaluation of trabecular bone patterns on dental radiographic images: influence of cortical bone”, dans *SPIE Medical Imaging 2010*, t. 7626, San Diego, États-Unis, fév. 2010, p. 76261M. DOI : 10.1117/12.843964.
- [3] Yves Amouriq, Jeanpierre Guédon, Nicolas Normand, **Aurore Arlicot**, Yasmine Benhdech, Pierre Weiss, “Bone texture analysis on dental radiographic images: results with several angulated radiographs on the same region of interest”, dans *SPIE Medical Imaging 2011*, t. 7965, Lake Buena Vista, Florida, États-Unis, fév. 2011, p. 796 525. DOI : 10.1117/12.878152.
- [7] **Aurore Arlicot**, Yves Amouriq, Pierre Evenou, Nicolas Normand, Jeanpierre Guédon, “A single scan skeletonization algorithm: application to medical imaging of trabecular bone”, dans *SPIE Medical Imaging 2010*, t. 7623, San Diego, CA, États-Unis, fév. 2010. DOI : 10.1117/12.844250.
- [8] **Aurore Arlicot**, Pierre Evenou, Nicolas Normand, “Single-scan skeletonization driven by a neighborhood-sequence distance”, dans *International Workshop on Combinatorial Image Analysis 2011*, Madrid : Research Publishing Service, mai 2011, p. 61–72, ISBN : 978-981-08-7923-5.

-
- [11] **Aurore Arlicot**, Bahman Soheilian, Nicolas Paparoditis, “Circular road sign extraction from street level images using colour, shape and texture database maps”, dans *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, t. 38, Paris, France, sept. 2009, p. 205–210.
 - [20] Pauline Bléry, Yves Amouriq, Jeanpierre Guédon, Paul Pilet, Nicolas Normand, Nicolas Durand, Florent Espitalier, **Aurore Arlicot**, Olivier Malard, Pierre Weiss, “Microarchitecture of irradiated bone: comparison with healthy bone”, dans *SPIE Medical Imaging 2012*, t. 8317, San Diego, États-Unis, fév. 2012. DOI : 10.1117/12.911375.
 - [60] Nicolas Normand, Robin Strand, Pierre Evenou, **Aurore Arlicot**, “Path-based distance with varying weights and neighborhood sequences”, dans *Discrete Geometry for Computing Imagery 2011*, sér. Lecture Notes in Computer Science, t. 6607, Nancy, France : Springer Berlin / Heidelberg, avr. 2011, p. 199–210. DOI : 10.1007/978-3-642-19867-0_17.
 - [84] Bahman Soheilian, **Aurore Arlicot**, Nicolas Paparoditis, “Extraction de panneaux de signalisation routière dans des images couleurs”, dans *Actes du 17^e congrès francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, Caen, France, jan. 2010.

Autres communications

- [1] Yves Amouriq, **Aurore Arlicot**, Pierre Evenou, Nicolas Normand, François Bodic, Pierre Layrolle, Jeanpierre Guédon, *Influence de l’os cortical dans l’appréciation du tissu osseux trabéculaire par radiographies retro alveolaires*. journées du Collège National des Enseignants en Prothèse Odontologique (CNEPO), Saint Malo, France, sept. 2009.
- [9] **Aurore Arlicot**, Nicolas Normand, Yves Amouriq, Jeanpierre Guédon, *Extraction of bone structure with a single-scan skeletonization driven by distance*, First Sino-French Workshop on Education and Research collaborations in Information and Communication Technologies, Nantes, France, mai 2011.

- [10] **Aurore Arlicot**, Nicolas Normand, Pierre Evenou, *Transformation d'un algorithme de squelettisation vers un calcul en un balayage*, Groupe de Travail de Géométrie Discrète, GDR Informatique Mathématique, Nantes, France, juin 2012.

Bibliographie

- [1] Yves Amouriq, **Aurore Arlicot**, Pierre Evenou, Nicolas Normand, François Bodic, Pierre Layrolle, Jeanpierre Guédon, *Influence de l'os cortical dans l'appréciation du tissu osseux trabéculaire par radiographies retro alveolaires*. journées du Collège National des Enseignants en Prothèse Odontologique (CNEPO), Saint Malo, France, sept. 2009.
- [2] Yves Amouriq, Pierre Evenou, **Aurore Arlicot**, Nicolas Normand, Pierre Layrolle, Pierre Weiss, Jeanpierre Guédon, “Evaluation of trabecular bone patterns on dental radiographic images: influence of cortical bone”, dans *SPIE Medical Imaging 2010*, t. 7626, San Diego, États-Unis, fév. 2010, p. 76261M. DOI : 10.1117/12.843964.
- [3] Yves Amouriq, Jeanpierre Guédon, Nicolas Normand, **Aurore Arlicot**, Yasmine Benhdech, Pierre Weiss, “Bone texture analysis on dental radiographic images: results with several angulated radiographs on the same region of interest”, dans *SPIE Medical Imaging 2011*, t. 7965, Lake Buena Vista, Florida, États-Unis, fév. 2011, p. 796525. DOI : 10.1117/12.878152.
- [4] Carlo Arcelli, “Pattern thinning by contour tracing”, *Computer Graphics and Image Processing*, t. 17, n° 2, p. 130 –144, 1981, ISSN : 0146-664X. DOI : 10.1016/0146-664X(81)90021-6.
- [5] Carlo Arcelli, Gabriella Sanniti di Baja, “A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, t. 11, n° 4, p. 411 –414, avr. 1989, ISSN : 0162-8828. DOI : 10.1109/34.19037.
- [6] Carlo Arcelli, Gabriella Sanniti di Baja, Luca Serino, “A parallel algorithm to skeletonize the distance transform of 3D objects”, *Image Vision Comput.*, t. 27, p. 666–672, 6 mai 2009, ISSN : 0262-8856. DOI : 10.1016/j.imavis.2008.07.014.

- [7] **Aurore Arlicot**, Yves Amouriq, Pierre Evenou, Nicolas Normand, Jean-pierre Guédon, “A single scan skeletonization algorithm: application to medical imaging of trabecular bone”, dans *SPIE Medical Imaging 2010*, t. 7623, San Diego, CA, États-Unis, fév. 2010. DOI : 10.1117/12.844250.
- [8] **Aurore Arlicot**, Pierre Evenou, Nicolas Normand, “Single-scan skeletonization driven by a neighborhood-sequence distance”, dans *International Workshop on Combinatorial Image Analysis 2011*, Madrid : Research Publishing Service, mai 2011, p. 61–72, ISBN : 978-981-08-7923-5.
- [9] **Aurore Arlicot**, Nicolas Normand, Yves Amouriq, Jeanpierre Guédon, *Extraction of bone structure with a single-scan skeletonization driven by distance*, First Sino-French Workshop on Education and Research collaborations in Information and Communication Technologies, Nantes, France, mai 2011.
- [10] **Aurore Arlicot**, Nicolas Normand, Pierre Evenou, *Transformation d’un algorithme de squelettisation vers un calcul en un balayage*, Groupe de Travail de Géométrie Discrète, GDR Informatique Mathématique, Nantes, France, juin 2012.
- [11] **Aurore Arlicot**, Bahman Soheilian, Nicolas Paparoditis, “Circular road sign extraction from street level images using colour, shape and texture database maps”, dans *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, t. 38, Paris, France, sept. 2009, p. 205–210.
- [12] Dominique Attali, Gabriella Sanniti di Baja, Édouard Thiel, “Skeleton simplification through non significant branch removal”, *Image Processing and Communications*, t. 3, n° 3-4, p. 63–72, 1997.
- [13] Dominique Attali, Edouard Thiel, “Du squelette discret ou continu”, dans *Discrete Geometry for Computing Imagery*, sér. Lecture Notes in Computer Science, Strasbourg, sept. 1993, p. 236–244.
- [14] G. Aufort, R. Jennane, R. Harba, C. L. Benhamou, “Hybrid skeleton graph analysis of disordered porous media. application to trabecular bone”, dans *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*.

- 2006 *IEEE International Conference on*, t. 2, mai 2006, p. II. DOI : 10 . 1109/ICASSP.2006.1660459.
- [15] G. Aufort, R. Jennane, R. Harba, A. Gasser, D. Soulat, C. L. Benhamou, “Nouvelle approche de modélisation de milieux poreux. application à l’os trabéculaire.”, dans *Groupe d’Etudes du Traitement du Signal et des Images*, 2005.
 - [16] Samuel Beatty, “Problem 3173”, *American Mathematical Monthly*, t. 33, n° 3, p. 159, mar. 1926. DOI : 10.2307/2300153.
 - [17] Samuel Beatty, A. Ostrowski, J. Hyslop, A. C. Aitken, “3177”, *American Mathematical Monthly*, t. 34, n° 3, pp. 159–160, 1927, ISSN : 00029890. DOI : 10.2307/2298716.
 - [18] C. L. Benhamou, L. Pothuau, E. Lespessailles, “Caractérisation de la microarchitecture trabéculaire osseuse par analyse de texture”, *ITBM-RBM*, t. 21, n° 6, p. 351 –358, 2000, ISSN : 1297-9562. DOI : 10 . 1016 /S1297 – 9562(00)90035–4.
 - [19] Gilles Bertrand, “Détermination de l’axe médian par sur-échantillonnage fictif”, dans *Reconnaissance des Formes et Intelligence Artificielle*, jan. 1984, p. 265–275.
 - [20] Pauline Bléry, Yves Amouriq, Jeanpierre Guédon, Paul Pilet, Nicolas Normand, Nicolas Durand, Florent Espitalier, **Aurore Arlicot**, Olivier Malard, Pierre Weiss, “Microarchitecture of irradiated bone: comparison with healthy bone”, dans *SPIE Medical Imaging 2012*, t. 8317, San Diego, États-Unis, fév. 2012. DOI : 10.1117/12.911375.
 - [21] Harry Blum, “A Transformation for Extracting New Descriptors of Shape”, dans *Models for the Perception of Speech and Visual Form*, Amsterdam : MIT Press, 1967, p. 362–380.
 - [22] Gunilla Borgefors, “Distance transformations in arbitrary dimensions”, *Computer Vision, Graphics, and Image Processing*, t. 27, n° 3, p. 321–345, sept. 1984. DOI : 10.1016/0734–189X(84)90035–5.

- [23] ———, “Weighted digital distance transforms in four dimensions”, *Discrete Applied Mathematics*, t. 125, n° 1, p. 161–176, jan. 2003. DOI : 10.1016/S0166-218X(02)00229-9.
- [24] Gunilla Borgefors, Sanniti di Baja Gabriella, “Skeletonizing the distance transform on the hexagonal grid”, dans *Pattern Recognition*, t. 1, nov. 1988, p. 504–507. DOI : 0.1109/ICPR.1988.28278.
- [25] Gunilla Borgefors, Ingela Nyström, Gabriella Sanniti di Baja, “Computing skeletons in three dimensions”, *Pattern Recognition*, t. 32, n° 7, p. 1225–1236, juil. 1999. DOI : 10.1016/S0031-3203(98)00082-X.
- [26] E. Brun, J. Vicente, F. Topin, R. Occelli, “Imorph : a 3D morphological tool to fully analyse all kind of cellular materials”, dans *Cellular Metals for Structural and Functional Applications*, 2008.
- [27] Odemir Martinez Bruno, Luciano Fontoura Costa, “A parallel implementation of exact euclidean distance transform based on exact dilations”, *Microprocessors and Microsystems*, t. 28, n° 3, p. 107–113, 2004, ISSN : 0141-9331. DOI : 10.1016/j.micpro.2004.01.001.
- [28] David W. Capson, Andrew Fung Chi-San, “Connected skeletons from 3D distance transforms”, dans *Image Analysis and Interpretation*, IEEE Southwest Symposium on, avr. 1998, p. 174–179. DOI : 10.1109/IAI.1998.666881.
- [29] Daniel Chappard, Erick Legrand, Christian Pascaretti, Michel F. Baslé, Maurice Audran, “Comparison of eight histomorphometric methods for measuring trabecular bone architecture by image analysis on histological sections”, *Microscopy Research and Technique*, t. 45, n° 4-5, p. 303–312, 1999, ISSN : 1097-0029. DOI : 10.1002/(SICI)1097-0029(19990515/01)45:4/5<303::AID-JEMT14>3.0.CO;2-8.
- [30] F. Chevalier, A. Laval-Jeantet, M. Laval-Jeantet, C. Bergot, “CT image analysis of the vertebral trabecular network in vivo”, *Calcified Tissue International*, t. 51, p. 8–13, 1 1992, ISSN : 0171-967X. DOI : 10.1007/BF00296208.
- [31] David Coeurjolly, Jérôme Hulin, Isabelle Sivignon, “Finding a minimum medial axis of a discrete shape is np-hard”, *Theor. Comput. Sci.*, t. 406, p. 72–79, 1-2 oct. 2008, ISSN : 0304-3975. DOI : 10.1016/j.tcs.2008.06.013.

-
- [32] Bernard Cortet, Xavier Marchandise, “Microarchitecture et résistance mécanique osseuse”, *Revue du Rhumatisme*, t. 68, n° 7, p. 584 –594, 2001, ISSN : 1169-8330. DOI : 10.1016/S1169-8330(01)00155-7.
- [33] Michel Couprie, “Note on fifteen 2D parallel thinning algorithms”, Université de Marne-la-Vallée, rap. tech., 2006.
- [34] L. Dalle Carbonare, M. T. Valenti, F. Bertoldo, M. Zanatta, S. Zenari, G. Realdi, V. Lo Cascio, S. Giannini, “Bone microarchitecture evaluated by histomorphometry”, *Micron*, t. 36, n° 7–8, p. 609 –616, 2005, ISSN : 0968-4328. DOI : 10.1016/j.micron.2005.07.007.
- [35] Per-Erik Danielsson, “Euclidean distance mapping”, *Computer Graphics and Image Processing*, t. 14, n° 3, p. 227 –248, 1980, ISSN : 0146-664X. DOI : 10.1016/0146-664X(80)90054-4.
- [36] P. P. Das, B. N. Chatterji, “Octagonal distances for digital pictures”, *Information Sciences*, t. 50, n° 2, p. 123–150, mar. 1990. DOI : 10.1016/0020-0255(90)90008-X.
- [37] Céline Fouard, Grégoire Malandain, “3-D chamfer distances and norms in anisotropic grids”, *Image and Vision Computing*, t. 23, n° 2, p. 143–158, fév. 2004. DOI : 10.1016/j.imavis.2004.06.009.
- [38] Céline Fouard, Grégoire Malandain, Steffen Prohaska, Malte Westerhoff, Francis Cassot, Christophe Mazel, Didier Asselot, Jean-Pierre Marc-Vergnes, “Squelettisation par blocs pour des grands volumes de données 3D: application à la microcirculation cérébrale”, dans *Proc. of RFIA’04*, Toulouse, France, jan. 2004.
- [39] Herbert Freeman, “On the encoding of arbitrary geometric configurations”, *IRE Trans. on Electronic Computers*, t. 10, n° 2, p. 260–268, juin 1961. DOI : 10.1109/TEC.1961.5219197.
- [40] Weixin Gong, “Nouveaux algorithmes de squelettisation en 2D et en 3D”, thèse de doct., École supérieure d’ingénieurs en électrotechnique et en électronique, 1992.

- [41] Stephen B. Gray, “Local properties of binary images in two dimensions”, *IEEE Transactions on Computers*, t. C-20, n° 5, p. 551–561, mai 1971, ISSN : 0018-9340. DOI : 10.1109/T-C.1971.223289.
- [42] Jeanpierre Guédon, Florent Autrusseau, Yves Amouriq, Pauline Bléry, Jean-Michel Bouler, Pierre Weiss, François-Xavier Barbarin, Thomas Dallet, Valentin Dallerit, “Exploring relationship between fractal dimension and trabecular bone characteristics”, dans *SPIE Medical Imaging*, t. 8317, San Diego, États-Unis, fév. 2012.
- [43] P Guggenbuhl, F Bodic, L Hamel, M F Baslé, D Chappard, “Texture analysis of X-ray radiographs of iliac bone is correlated with bone micro-CT”, *Osteoporos Int*, t. 17, n° 3, p. 447–454, mar. 2006. DOI : 10.1007/s00198-005-0007-8.
- [44] András Hajdu, Lajos Hajdu, “Approximating the Euclidean distance using non-periodic neighbourhood sequences”, *Discrete Mathematics*, t. 283, n° 1-3, p. 101–111, juin 2004. DOI : 10.1016/j.disc.2003.12.016.
- [45] C. J. Hilditch, “Linear skeletons from square cupboards”, dans *Machine Intelligence*, cité dans [48], t. 4, 1969, p. 403–420.
- [46] Yan Hua, Olivia Nackaerts, Joke Duyck, Frederik Maes, Reinhilde Jacobs, “Bone quality assessment based on cone beam computed tomography imaging”, *Clinical Oral Implants Research*, t. 20, n° 8, p. 767–771, 2009, ISSN : 1600-0501. DOI : 10.1111/j.1600-0501.2008.01677.x.
- [47] Reinhard Klette, Azriel Rosenfeld, “Digital straightness—a review”, *Discrete Applied Mathematics*, t. 139, n° 1-3, p. 197–230, avr. 2004. DOI : 10.1016/j.dam.2002.12.001.
- [48] T. Y. Kong, Azriel Rosenfeld, “Digital topology: introduction and survey”, *Computer Vision, Graphics, and Image Processing*, t. 48, n° 3, p. 357–393, déc. 1989. DOI : 10.1016/0734-189X(89)90147-3.
- [49] Joachim Lambek, Leo Moser, “Inverse and complementary sequences of natural numbers”, *American Mathematical Monthly*, t. 61, n° 7, p. 454–458, août 1954, ISSN : 00029890. DOI : 10.2307/2308078.

-
- [50] Louisa Lam, Seong-Whan Lee, Ching Y. Suen, “Thinning methodologies-a comprehensive survey”, *Pattern Analysis and Machine Intelligence*, t. 14, p. 869–885, 9 sept. 1992, ISSN : 0162-8828. DOI : 10.1109/34.161346.
- [51] Christophe Lohou, “Contribution à l’analyse topologique des images : étude d’algorithmes de squelettisation pour images 2D et 3D, selon une approche topologique digitale ou topologique discrète”, thèse de doct., Université de Marne-la-Vallée, déc. 2001.
- [52] Christophe Lohou, Gilles Bertrand, “A 3D 6-subiteration curve thinning algorithm based on P-simple points”, *Discrete Applied Mathematics*, t. 151, n° 1-3, p. 198–228, 2005, ISSN : 0166-218X. DOI : <http://dx.doi.org/10.1016/j.dam.2005.02.030>.
- [53] Serge Miguet, Virginie Marion-Poty, “A new 2-D and 3-D thinning algorithm based on successive border generations”, dans *Discrete Geometry for Computing Imagery*, Grenoble, sept. 1994, p. 195–206.
- [54] Ugo Montanari, “A method for obtaining skeletons using a quasi-Euclidean distance”, *Journal of ACM*, t. 15, n° 4, p. 600–624, oct. 1968. DOI : 10.1145/321479.321486.
- [55] Annick Montanvert, “Contribution au traitement de formes discrètes : squelettes et codage par graphe de la ligne médiane.”, thèse de doct., Université de Scientifique, Technologique et Médicale de Grenoble, oct. 1987.
- [56] Benedek Nagy, “Distance with generalized neighbourhood sequences in nD and ∞D ”, *Discrete Applied Mathematics*, t. 156, n° 12, p. 2344–2351, 2008. DOI : 10.1016/j.dam.2007.10.017.
- [57] Nicolas Normand, “Représentation d’images et distances discrètes basées sur les éléments structurants à deux pixels”, thèse de doct., Université de Nantes, Nantes, France, jan. 1997.
- [58] Nicolas Normand, Robin Strand, Pierre Evenou, **Aurore Arlicot**, “A minimal delay neighborhood-sequence distance transform”, *Image Processing On Line*, 2012, en cours de révision.

- [59] ———, “Minimal-delay distance transform for neighborhood-sequence distances in 2D and 3D”, *Computer Vision and Image Understanding*, 2012, en cours de révision.
- [60] ———, “Path-based distance with varying weights and neighborhood sequences”, dans *Discrete Geometry for Computing Imagery 2011*, sér. Lecture Notes in Computer Science, t. 6607, Nancy, France : Springer Berlin / Heidelberg, avr. 2011, p. 199–210. DOI : 10.1007/978-3-642-19867-0_17.
- [61] A. Odgaard, “Three-dimensional methods for quantification of cancellous bone architecture”, *Bone*, t. 20, n° 4, p. 315 –328, 1997, ISSN : 8756-3282. DOI : 10.1016/S8756-3282(97)00007-0.
- [62] R. L. Ogniewicz, O. Kubler, “Hierarchic Voronoi skeletons”, *Pattern Recognition*, t. 28, n° 3, p. 343 –359, 1995, ISSN : 0031-3203. DOI : 10.1016/0031-3203(94)00105-U.
- [63] Kálmán Palágyi, “A 3-subiteration surface-thinning algorithm”, dans *Proceedings of the 12th international conference on Computer analysis of images and patterns*, sér. CAIP’07, Vienna, Austria : Springer-Verlag, 2007, p. 628–635, ISBN : 978-3-540-74271-5.
- [64] Kálmán Palágyi, Erich Sorantin, Emese Balogh, Attila Kuba, Csongor Halmai, Balázs Erdohelyi, Klaus Hasegger, “A sequential 3D thinning algorithm and its medical applications”, dans *IPMI ’01: Proceedings of the 17th International Conference on Information Processing in Medical Imaging*, London, UK : Springer-Verlag, 2001, p. 409–415, ISBN : 3-540-42245-5.
- [65] A. Michael Parfitt, Marc K. Drezner, Francis H. Glorieux, John A. Kanis, Hartmut Malluche, Pierre J. Meunier, Susan M. Ott, Robert R. Recker, “Bone histomorphometry: standardization of nomenclature, symbols, and units: report of the asbmr histomorphometry nomenclature committee”, *Journal of Bone and Mineral Research*, t. 2, n° 6, p. 595–610, 1987, ISSN : 1523-4681. DOI : 10.1002/jbmr.5650020617.
- [66] A M Parfitt, C H Mathews, A R Villanueva, M Kleerekoper, B Frame, D S Rao, “Relationships between surface, volume, and thickness of iliac trabecular bone in aging and in osteoporosis. implications for the microanatomic

- and cellular mechanisms of bone loss”, *The Journal of Clinical Investigation*, t. 72, n° 4, p. 1396–1409, oct. 1983. DOI : 10.1172/JCI111096.
- [67] Theo Pavlidis, “A thinning algorithm for discrete binary images”, *Computer Graphics and Image Processing*, t. 13, n° 2, p. 142 –157, 1980, ISSN : 0146-664X. DOI : 10.1016/S0146-664X(80)80037-2.
- [68] Françoise Peyrin, “Point sur les perspectives technologiques d’analyse de la microarchitecture osseuse”, *ITBM-RBM*, t. 22, n° 1, p. 5 –14, 2001, ISSN : 1297-9562. DOI : 10.1016/S1297-9562(01)90041-5.
- [69] Françoise Peyrin, Peter Zsolt, Aymeric Larrue, Alexandra Bonnassie, Dominique Attali, “Local geometrical analysis of 3d porous networks based on the medial axis: application to bone microarchitecture microtomography images”, *Image Analysis and Stereology*, t. 26, n° 3, 2011, ISSN : 1854-5165.
- [70] Laurent Pothuaud, Bert Van Rietbergen, Cindy Charlot, Eugene Ozhinsky, Sharmila Majumdar, “A new computational efficient approach for trabecular bone analysis using beam models generated with skeletonized graph technique”, *Computer Methods in Biomechanics and Biomedical Engineering*, t. 7, n° 4, p. 205–213, 2004. DOI : 10.1080/10255840412331285943.
- [71] Chris Pudney, “Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images”, *Computer Vision and Image Understanding*, t. 72, n° 3, p. 404–413, 1998, ISSN : 1077-3142. DOI : <http://dx.doi.org/10.1006/cviu.1998.0680>.
- [72] Ingemar Ragnemalm, “The euclidean distance transform in arbitrary dimensions”, dans *Image Processing and its Applications, 1992., International Conference on*, avr. 1992, p. 290 –293. DOI : 10.1016/0167-8655(93)90152-4.
- [73] Alain Ramé, Sylvie Théron, *Anatomie et physiologie*. Elsevier Masson, 2006, ISBN : 2842998340.
- [74] Vincent Ranwez, Pierre Soille, “Order independent homotopic thinning”, dans *Discrete Geometry for Computing Imagery*, sér. Lecture Notes in Computer Science, London, UK : Springer-Verlag, 1999, p. 337–348, ISBN : 3-540-65685-5.

- [75] Éric Rémy, “Normes de chanfrein et axe médian dans le volume discret”, thèse de doct., Université de la Méditerranée, 17 déc. 2001.
- [76] Éric Rémy, Édouard Thiel, “Optimizing 3D chamfer masks with norm constraints”, dans *International Workshop on Combinatorial Image Analysis*, Caen, juil. 2000, p. 39–56.
- [77] Azriel Rosenfeld, “Arcs and curves in digital pictures”, *Journal of ACM*, t. 20, n° 1, p. 81–87, jan. 1973.
- [78] Azriel Rosenfeld, John L. Pfaltz, “Distances functions on digital pictures”, *Pattern Recognition*, t. 1, n° 1, p. 33–61, juil. 1968. DOI : 10.1016/0031-3203(68)90013-7.
- [79] —, “Sequential operations in digital picture processing”, *Journal of ACM*, t. 13, n° 4, p. 471–494, oct. 1966, ISSN : 0004-5411. DOI : 10.1145/321356.321357.
- [80] Denis Rutovitz, “Pattern recognition”, dans *Journal of Royal Statistical Society*, 1966, p. 504–530.
- [81] Gabriella. Sanniti di Baja, Édouard Thiel, “Computing and comparing distance-driven skeletons”, dans *Aspects of Visual Form Processing*, World Scientific, Singapore, 1994, p. 465–486.
- [82] J. A. Sethian, “Curvature flow and entropy conditions applied to grid generation”, *Journal of Computational Physics*, t. 115, n° 2, p. 440–454, 1994, ISSN : 0021-9991. DOI : 10.1006/jcph.1994.1209.
- [83] —, “Fast marching methods”, *SIAM Rev.*, t. 41, p. 199–235, 2 juin 1999, ISSN : 0036-1445. DOI : <http://dx.doi.org/10.1137/S0036144598347059>.
- [84] Bahman Soheilian, **Aurore Arlicot**, Nicolas Paparoditis, “Extraction de panneaux de signalisation routière dans des images couleurs”, dans *Actes du 17^e congrès francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, Caen, France, jan. 2010.
- [85] Robin Strand, Benedek Nagy, “A weighted neighbourhood sequence distance function with three local steps”, dans *7th Int’l Symposium on Image and Signal Processing and Analysis*, Dubrovnik, Croatia, sept. 2011, p. 564–568.

-
- [86] ———, “Distances based on neighbourhood sequences in non-standard three-dimensional grids”, *Discrete Applied Mathematics*, t. 155, n° 4, p. 548–557, fév. 2007. DOI : 10.1016/j.dam.2006.09.005.
- [87] *Structural parameters measured by skyscan CT-analyser software*, SKYSCAN, Kontich, Belgium, août 2009.
- [88] Stina Svensson, Gunilla Borgefors, “Digital distance transforms in 3d images using information from neighbourhoods up to $5 \times 5 \times 5$ ”, *Computer Vision and Image Understanding*, t. 88, n° 1, p. 24–53, oct. 2002. DOI : 10.1006/cviu.2002.0976.
- [89] ———, “Distance transforms in 3D using four different weights”, *Pattern Recognition Letter*, t. 23, n° 12, p. 1407–1418, oct. 2002. DOI : 10.1016/S0167-8655(02)00101-0.
- [90] Stina Svensson, Gunilla Borgefors, Ingela Nyström, “On reversible skeletonization using anchor-points from distance transforms”, *Journal of Visual Communication and Image Representation*, t. 10, n° 4, p. 379–397, 1999, ISSN : 1047-3203. DOI : 10.1006/jvci.1999.0425.
- [91] A. Taleb-Ahmed, P. Dubois, E. Duquenoy, “Analysis methods of CT-scan images for the characterization of the bone texture: first results”, *Pattern Recognition Letter*, t. 24, p. 1971–1982, 12 août 2003, ISSN : 0167-8655. DOI : [http://dx.doi.org/10.1016/S0167-8655\(03\)00036-9](http://dx.doi.org/10.1016/S0167-8655(03)00036-9).
- [92] Edouard Thiel, “Unification de la squelettisation menée en distance”, dans *9^e Reconnaissance des Formes et Intelligence Artificielle*, t. 1, Paris, jan. 1994, p. 349–358.
- [93] Édouard Thiel, “Géométrie des distances de chanfrein”, HDR, Université de la Méditerranée, 10 déc. 2001.
- [94] ———, “Les distances de chanfrein en analyse d’images : fondements et applications”, <http://www.lif-sud.univ-mrs.fr/~thiel/these>, Thèse de doctorat, Université Joseph Fourier, Grenoble 1, sept. 1994.

- [95] Ben J. H. Verwer, Piet W. Verbeek, Simon T. Dekker, “An efficient uniform cost algorithm applied to distance transforms”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, t. 11, n° 4, p. 425–429, avr. 1989. DOI : 10.1109/34.19041.
- [96] Lucas J. Vliet, Ben J. H. Verwer, “A contour processing method for fast binary neighbourhood operations”, *Pattern Recognition Letter*, t. 7, n° 1, p. 27–36, avr. 1984, ISSN : 0167-8655. DOI : 10.1016/0167-8655(88)90041-4.
- [97] Xiaoli Wang, Gilles Bertrand, “Some sequential algorithms for a generalized distance transformation based on Minkowski operations”, *Pattern Analysis and Machine Intelligence*, t. 14, n° 11, p. 1114–1121, nov. 1992, ISSN : 0162-8828. DOI : 10.1109/34.166628.
- [98] Shigeki Yokoi, Jun-ichiro Toriwaki, Teruo Fukumura, “An analysis of topological properties of digitized binary pictures using local features”, *Computer Graphics and Image Processing*, t. 4, n° 1, p. 63–73, 1975, ISSN : 0146-664X. DOI : 10.1016/0146-664X(75)90022-2.
- [99] Y. Y. Zhang, P. S. P. Wang, “A new parallel thinning methodology”, *Int. J. Pattern Recognition Artif. Intell.*, t. 8, n° 5, p. 999–1011, 1994.

Thèse de Doctorat

Aurore Arlicot

Squelettisation en un balayage.

Application à la caractérisation osseuse.

Résumé

La squelettisation est une méthode de reconnaissance et de caractérisation des formes utilisant des notions de topologie discrète, qui permet d'extraire le squelette d'une forme binaire. Ce squelette (un sous ensemble de la forme centré, fin, topologiquement équivalent à la forme et restructurable) se calcule grâce à plusieurs balayages de l'image. Dans ce travail de thèse nous nous sommes intéressés au calcul du squelette en une seule passe (mode « streaming »). Pour cela nous avons proposé un nouvel algorithme qui ré-ordonne les calculs sur chaque pixel dans leur ordre d'apparition dans l'image (sans changer le sens de propagation de l'information dans l'image) en calculant une carte de distance asymétrique puis qui trouve les points de l'axe médian. Selon un ordre particulier, qui garantit l'identité de nos résultats avec ceux d'un algorithme de référence, nous pouvons supprimer les points de la carte de distance en ne conservant que les points du squelette dont les points de l'axe médian. En plus du fait que nos squelettes soient identiques au pixel près à ceux obtenus par l'algorithme de référence, notre algorithme les calcule plus rapidement. Bien que notre approche soit générique, nous avons utilisé cet algorithme pour développer un outil logiciel en imagerie médicale permettant d'extraire les caractéristiques osseuses dans le but de mesurer la structure osseuse à partir du squelette obtenu sur des images de micro-scanner. L'extraction de zones d'os trabéculaire 2D sur des images binarisées, permet bien une caractérisation de l'os en accord avec la vérité terrain obtenue par d'autres logiciels.

Mots clés

squelettisation, streaming, os trabéculaire, micro-architecture osseuse.

Abstract

Skeletonization is a method for shape recognition and characterization that uses discrete geometry notions. Given a binary image, this method allows the extraction of a skeleton. This skeleton is computed thanks to several scans of the image. It is a centered subset of the shape, thin, topologically equivalent to the input shape and it can be reconstructed. In this thesis we aimed at computing the skeleton with only one scan (streaming mode). We thus proposed a new algorithm that re-organizes the computation on each pixel according to their occurrence order within the image in order to compute an asymmetric distance map and to find the medial axis points. The method does not change the direction of the image information propagation. According to a particular order, which guarantees the same results than those of the reference algorithm results, we can delete the points from the asymmetric distance map while conserving the skeletal points and the medial axis points. Besides, the fact that our skeletons are identical to the pixel to those obtained by the reference method, our algorithm is the fastest. Although our approach is generic, we used this algorithm to develop a software tool for medical imaging applications. Trabecular bones characteristics extraction from 2D binarized images enables bones characterization that matches the observations yielded from other software.

Key Words

skeletonization, streaming, trabecular bone, bone microarchitecture.